Master of Science in Artificial Intelligence-MSAI

AI6102: Machine Learning Methodologies Applications

Group Project Kaggle Competition

Project Name: Multi-Class Prediction of Obesity Risk

Team Members

Names: Maheep[1], Ho Chek Hui[2], Ntambara Etienne[3]

Matriculation Number: G2303665G[1], G2303025C[2], G2304253K[3]

Computer Science and Engineering
Nanyang Technological University
Singapore
April 30, 2024

Project Report: Multi-Class Prediction of Obesity Risk

Maheep[1], Ho Chek Hui[2], Ntambara Etienne[3]
Nanyang Technological University
Singapore
maheep001@e.ntu.edu.sg
S230075@e.ntu.edu.sg
ntam0001@e.ntu.edu.sg

Figure 1: Competition Leaderbaord

1 Introduction

In this report, we described the approaches to tackle an ongoing Kaggle Competition titled 'Multi-Class Prediction of Obesity Risk'. By performing in-depth exploratory data analysis and using insights gained to engineer features and tune models, our team was able to achieve a top 7% in the leaderboard, clinching 238th place out of 3,587 teams globally. The obtained results are visible in Figure 1.

2 Problem Statement

This Kaggle competition (Multi-Class Prediction of Obesity Risk) [3] requires participants to perform multi-class classification on structured data to predict an individual's risk of obesity based on a variety of input factors. Features include both continuous and categorical data such as height, gender, consumption of alcohol, etc. Submissions are evaluated based on accuracy of the test dataset.

3 Challenges of the problem

There are 2 main challenges identified:

- Limited information on how the data is derived, thus preventing us from acquiring or synthesizing more data.
- High Baseline Performance and Leaderboard Saturation. A simple baseline can easily obtain 90% accuracy, so improving the model to attain the top 10% in the leaderboard from an already high baseline is difficult.

4 Proposed Solution

This section is dedicated to showcase the different procedures used to accomplish the solution of the above problems. Furthermore, we discuss these procedures in detail including Exploratory Data Analysis (EDA), Feature Engineering, Cross-Validation, Modelling and Hyperparameter Tuning, and Ensemble Technique. At the end, we also discuss about the Hill Climbing Optimization.

4.1 Exploratory Data Analysis (EDA)

Exploratory data analysis will be performed to understand the need for preprocessing and gain insights on the data for feature engineering. Some of the questions that would be better understood during this phase include:

1. Distribution of continuous and categorical features
2. Proportion of data in each target class
3. Presence of null values or outliers
4. Features that are correlated with one another or with the target variable

4.2 Feature Engineering

Feature Engineering refers to the process of transforming the existing features into new features that are more relevant for the task. To gain insight about the transformation, researchers use Exploratory Data Analysis and other processes. The data is transformed such that the representation of the features or the presence of salient feature increases, while reducing the presence of redundant features. Some of the example of feature engineering are dimensionality reduction using PCA, or handling missing values, etc. We will explore the usage of feature engineering in depth in Section 5.3.

4.3 Cross-Validation

To prevent overfitting, a validation dataset is used to verify the performance of the machine learning model on unseen data. K-fold stratified cross-validation is used to preserve the proportion of the data while ensuring the entire dataset is utilized to fit the model across different folds. A higher K-fold is selected as it reduces the number of data instances in each fold, and given that 1 fold serves as the validation dataset, the remaining training dataset contains a larger number of data for fitting the model. The cross-validation score will be used instead of the public leaderboard score, to determine which submission is used for the final evaluation in the private leaderboard, since the former is calculated based on the entire training dataset which has ~20,000 rows while the public leaderboard is based on ~2,700 rows. As such, the cross-validation score will be more robust to noise and outliers due to the larger amount of data.

4.4 Modelling and Hyperparameter Tuning

Given that the optimal model depends on the data, different models are used to investigate which model produces the best results. Predictions made by different models can be combined via an ensemble approach. Using different models increases diversity and allows errors made by a model to be mitigated by another model. In particular, tree-based algorithms are preferred as it has shown good results with tabular data and outperformed neural networks [2]. However, a particular neural-network model called TabNet will be used since it was reported to work well for previous Kaggle competitions [1]. Hyperparamater tuning will be done via Bayesian Optimization using a framework called Optuna. This allows one to optimize the hyperparameters automatically in an efficient manner without having to try out every hyperparameters defined within the search space.

4.5 Ensemble

Predictions made by the model are combined by taking a simple average over the probabilities of each model. Label of the data row is assigned based on the class with the highest probabilities. While it is possible to use Bayesian Optmization to determine the optimal weightage of each model, it might inevitably lead to over-fitting on the validation dataset, and as such a simple averaging across the model predictions will be used instead.

A hill-climbing optimization approach is used for the selection of models where we start with an initial model and add a new model to the ensemble mix one at a time. If the introduction of the new model improves the overall cross-validation score, then it is kept. However, if the introduction decreases the score then we discard the model and introduce a different model to the mix. The pseudo-code below shows how the optimization will be done.

Algorithm 1 Hill-Climbing Optimization

Require: $number\_of\_models > 0$
  $current\_ensemble \leftarrow initial\_model$
  $current\_score \leftarrow evaluate(current\_ensemble)$
  while $number\_of\_models > 0$ **do**
    $new\_ensemble \leftarrow current\_ensemble + new\_model$
    $new\_score \leftarrow evaluate(new\_ensemble)$
    if $new\_score >= current\_score$ **then**
      $current\_ensemble \leftarrow new\_ensemble$
      $current\_score \leftarrow new\_score$
    end if
    $number\_of\_models \leftarrow number\_of\_models - 1$
  end while
  return $current\_ensemble$

5 Experimentation

In this section, we discuss the approaches used to implement the stated process in the Proposed Work Section 4. We state the different approaches and models used to achieve the accuracy on the dataset.

5.1 Experiment Set-up

All the experiments are done using CPU on Kaggle platform. Experiment runs are tracked using an MLOps platform called 'Weights and Biases' which stores the hyperparameters used, performance of the model, and all the logs generated from each run.

5.2 Exploratory Data Analysis (EDA)

To understand the distribution of the data and identify any outliers, univariate data analysis is performed for each feature. Figure 2 shows the distribution of the continuous features while Figure 3 shows the same for the categorical features. Some observations include 'Age' being skewed to the left indicating a generally young population with outliers above 35 years old. Distribution of Weight is wide ranging from 40kg to 160kg while the distribution of Height is more symmetrical.

Figure 2: Distribution of continuous features

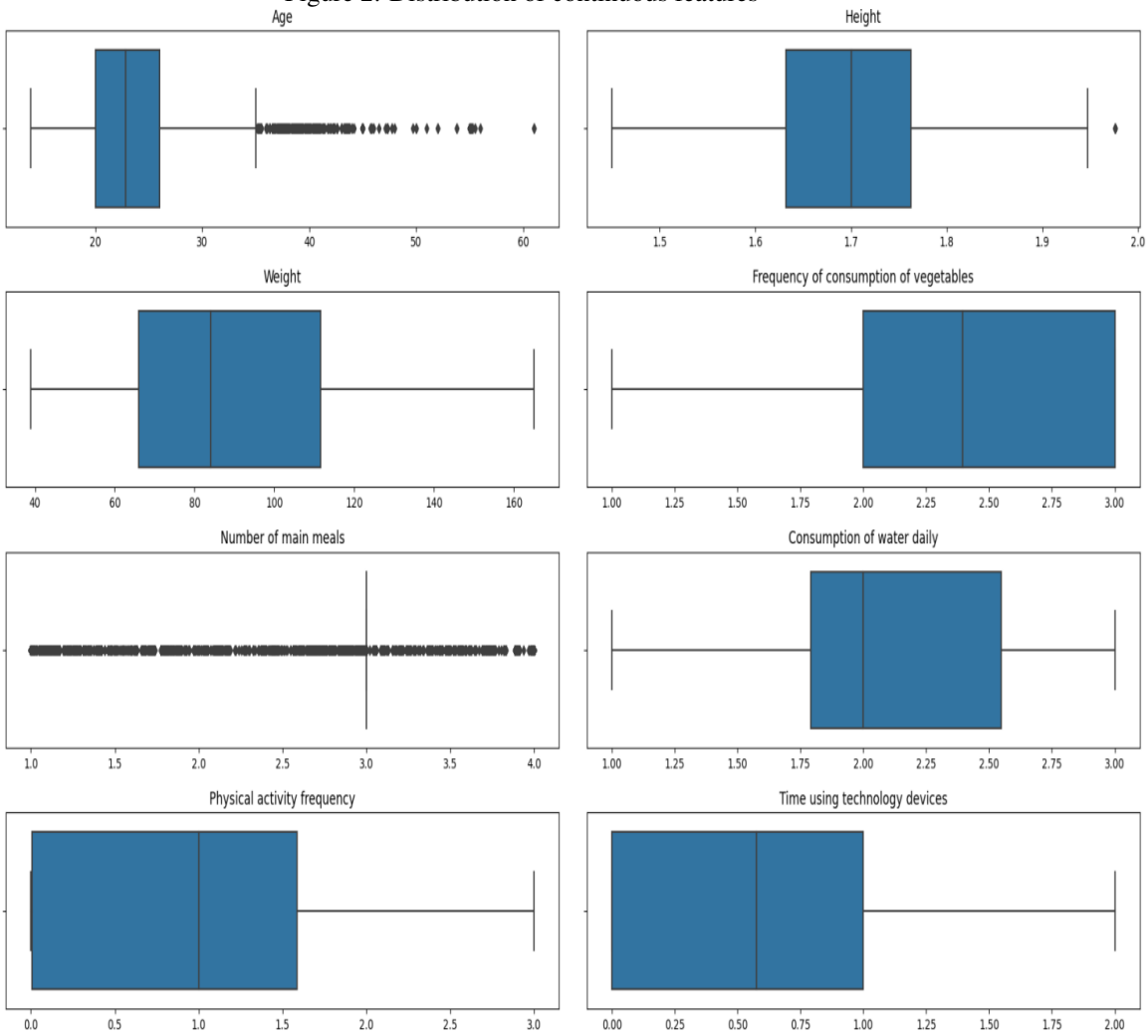
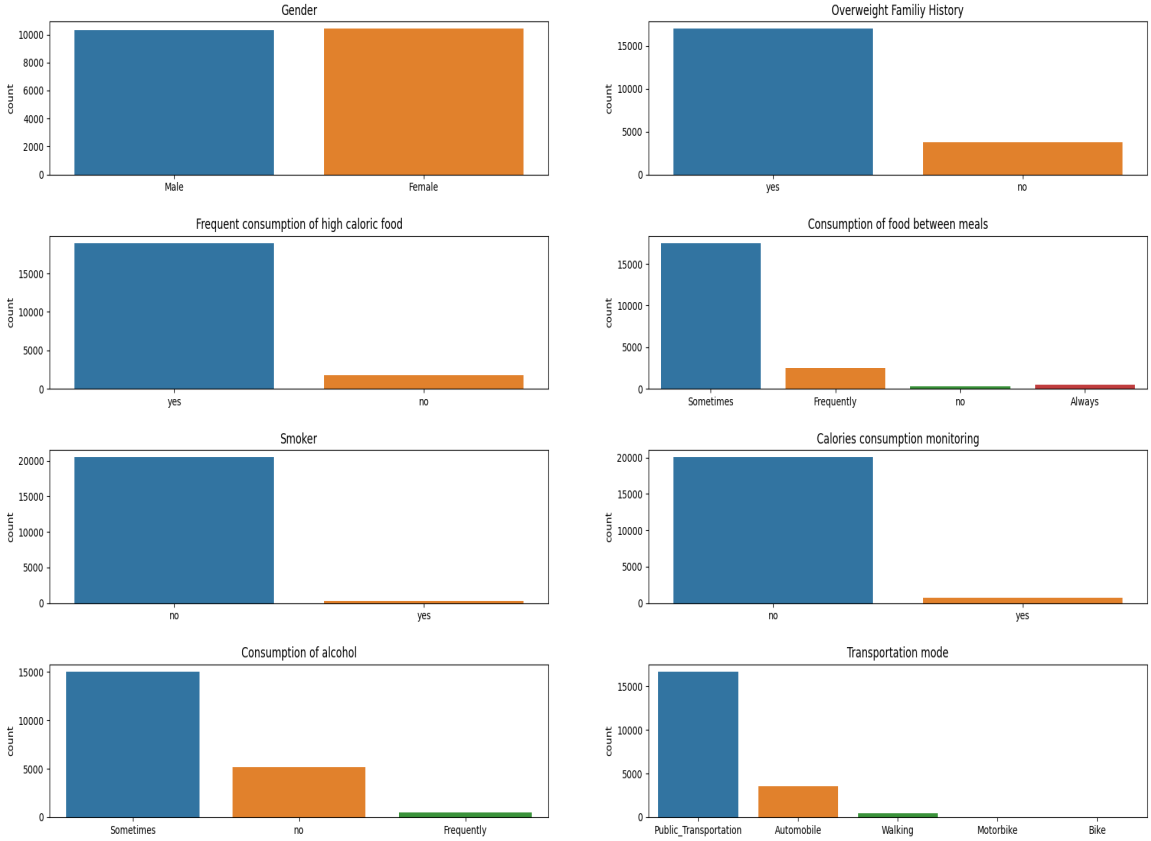
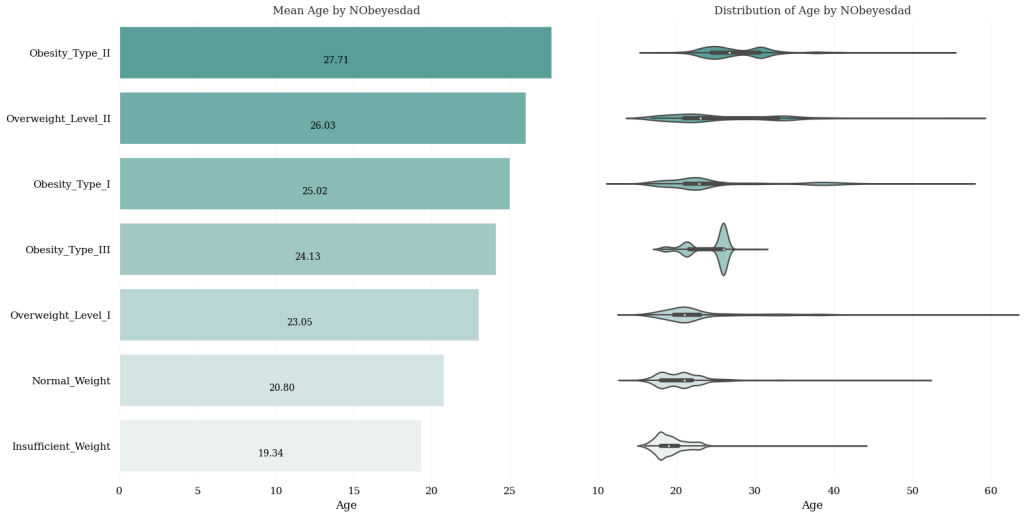


Figure 3: Distribution of categorical features



Bivariate data analysis is also performed to investigate which features are correlated with the target variables. An example is shown in Figure 4 where we observed Age to be correlated with a higher risk of obesity.

Figure 4: Age and Target variable



5.3 Feature Engineering

Using the insights gained from exploratory data analysis and contextual knowledge of obesity, several new features are created and are shown in Table 1.

S/N	Description	Formula
1	BMI	$(\text{Height}/\text{Weight}^2)$
2	BMI Grouping	Underweight if BMI < 18.5 etc.
3	BMI and Number of Meals Consumed	BMI * Number of Meals Consumed
4	BMI and Physical activity frequency	BMI * Physical activity frequency
5	Physical activity frequency and time using technology devices	Physical activity frequency - time using technology devices
6	Frequency of consuming vegetables and Number of Meals Consumed	Frequency of consuming vegetables * Number of Meals Consumed

Table 1: New features

5.4 Modelling

An initial approach was implemented to establish a baseline result. This approach uses all the raw and derived features with a 10-fold cross-validation. Hyperparameters were tuned with Optuna and accuracy was used as the objective metric. Different tree-based models using scikit-learn library were used and the results in Table 2 show the Cross-Validation (CV) score and the public leaderboard (LB) score.

Version	Model	CV	LB
1	XGBoost	0.9114	0.90895
2	LightGBM	0.9115	0.9057
3	Catboost	0.9078	0.90317
4	Random Forest	0.8953	0.89161

Table 2: Baseline Results

After establishing a baseline, different tweaks were tried out to improve the results and also generate more diverse models to be used for ensemble. One of the tweaks performed was to use F1 score as an objective metric instead of accuracy during the hyperparameter tuning phase. This is simple to implement but allows the optimization process to be done differently. Table 3 shows the results from this approach.

Version	Model	CV	LB
5	XGBoost	0.9114	0.90968
6	LightGBM	0.9104	0.90715
7	Catboost	0.9082	0.9039
8	Random Forest	0.8955	0.89378

Table 3: F1 score as an optimizing metric

Apart from a slight improvement of CV score for CatBoost and Random Forest, the previous approach did not improve the baseline results much. The next approach retains the methodology in the baseline approach but uses additional data from another dataset. The dataset provided in this Kaggle competition was derived using another dataset hence the original dataset was added to the training data in this approach. In addition, a new model called TabNet that uses Attention mechanism is implemented. Results are shown in Table 4.

Apart from XGBoost, which is the best-performing model across all the different runs, introducing more data did not change the baseline results much. TabNet underperforms compared to other models in this dataset with a ~3% difference. As such, to reduce compute costs, TabNet is not used for further runs. Given that the dataset is relatively small, using more features could result in overfitting. Hence, in the next approach shown in Table 5, only the original features plus BMI-related features are used.

Version	Model	CV	LB
9	XGBoost	0.9125	0.90859
10	LightGBM	0.9106	0.9057
11	Catboost	0.9079	0.90354
12	Random Forest	0.8943	0.89234
13	TabNet	0.8833	0.8742

Table 4: Supplementing with additional data

Version	Model	CV	LB
14	CatBoost	0.9082	0.9104
15	LightGBM	0.9102	0.91943
16	Random Forest	0.9017	0.89812
17	XGBoost	0.9124	0.91582

Table 5: Using original and BMI-related features

In the final approach, only XGBoost and LightGBM are used as they produces the best result for both CV and LB across all the different runs. The number of K-folds was increased to 15 so the size of the training data increases. Results are shown in Table 6.

Version	Model	CV	LB
18	LightGBM	0.9122	0.90462
19	XGBoost	0.9144	0.90859

Table 6: 15 Fold Cross-Validation

5.5 Ensemble

With all the models trained across different experiment runs, an ensemble approach is used to combine the model predictions. Hill-climbing optimization approach is used to choose the models for the submission and Table 7 below shows how the final submission is derived by including a new model one at a time. The final submission consists of 4 XGBoost, 2 LightGBM, and 1 CatBoost.

Models	CV	LB
(1), (2)	0.91111	0.9072
(1), (2), (5)	0.91112	0.90859
(1), (2), (5), (14)	0.91256	0.9086
(1), (2), (5), (14), (15)	0.91314	0.9144
(1), (2), (5), (14), (15), (17)	0.91367	0.91473
(1), (2), (5), (14), (15), (17), (19)	0.91391	0.9147

Table 7: Submission: Hill-Climbing Ensemble Approach

Our final submission has a CV score of 0.91391, public leaderboard score of 0.9147 and a private leaderboard score of 0.90986. The submission was placed in the top 7% among 3,587 teams as shown in Figure 1.

6 Conclusion

In this report, we outlined our approach to tackling the Kaggle competition - 'Multi-Class Prediction of Obesity Risk' and described all the experimentation done. By performing an exploratory data exploration, we were able to gain useful insights on how to approach feature engineering. Features created were used for the building of machine learning models and by using an ensemble approach, our team was able to iteratively improve on the overall performance culminating in achieving a top 7% in the competition.

References

[1] ARIK, S. O., AND PFISTER, T. Tabnet: Attentive interpretable tabular learning. https://arxiv.org/abs/1908.07442, 2019.

[2] GRINSZTAJN, L., OYALLON, E., AND VAROQUAUX, G. Why do tree-based models still outperform deep learning on tabular data? https://arxiv.org/abs/2207.08815, 2022.

[3] READE, W., AND CHOW, A. Multi-class prediction of obesity risk. https://kaggle.com/competitions/playground-series-s4e2, 2024.