Multi-Class Classification of Obesity Risk

Report prepared by Aparna K

Introduction

The Kaggle project challenged participants to use various factors to predict obesity risk in individuals, which is related to cardiovascular disease. This report summarizes my approach to solving this challenge, where I achieved a submission score of **90.8**, close to the highest leaderboard score of **91.1**. This notable performance underscores my expertise in machine learning and data science providing a balanced opportunity to showcase data preprocessing, feature engineering, model tuning, and evaluation techniques, emphasizing practical skills in handling real-world datasets.

Dataset Overview

The data consist of the estimation of obesity levels in people with ages between 14 and 61 and diverse eating habits and physical condition.

The attributes related with eating habits are:

1. Frequent consumption of high caloric food (FAVC)

2. Frequency of consumption of vegetables (FCVC)

3. Number of main meals (NCP)

4. Consumption of food between meals (CAEC)

5. Consumption of water daily (CH20)

6. Consumption of alcohol (CALC)

The attributes related with the physical condition are:

7. Calories consumption monitoring (SCC)

8. Physical activity frequency (FAF)

9. Time using technology devices (TUE)

10. Transportation used (MTRANS)

Other variables include Gender, Age, Height and Weight.

Target variable ('Nobeyesdad') contains of the following categorical values - Insufficient_Weight, Normal_Weight, Obesity_Type_I, Obesity_Type_II, Obesity_Type_III, Overweight_Level_I, Overweight_Level_II

Exploratory Data Analysis (EDA)

During data cleaning and preprocessing, the 'CALC' column's category 'Always' was replaced with 'Frequently' to simplify the analysis, and the 'id' column was dropped as it was not relevant and could potentially harm the results.

Descriptive statistics were generated to understand the distribution of numerical features, and it was confirmed that there were no missing values in the dataset.

For feature engineering, the number of unique values in each column was printed to identify categorical features. Label encoding was performed on categorical columns such as 'Gender', 'family_history_with_overweight', 'FAVC', 'SMOKE', and 'SCC'. The target variable 'NObeyesdad' was also label encoded.

One-hot encoding was applied to the categorical columns 'MTRANS', 'CAEC', and 'CALC' to convert them into numerical format.

It was observed that there was no significant class imbalance in the output feature 'NObeyesdad'

Finally, the dataset was split into training and testing sets using an 80-20 split

Models Used

1. Random Forest Classifier

A Random Forest Classifier was trained to predict the target variable using the competition dataset. The model was initialized with 100 estimators and to evaluate its performance, a 5-fold cross-validation approach was employed, generating accuracy scores that provided insights into the model's consistency across different subsets of the training data. After fitting the model on the entire training dataset, predictions were made on the test set. The model achieved an accuracy of **89.52%** and a weighted F1 score of **89.52%.**

2. Neural Network

A neural network model was constructed and trained using Keras to predict the target variable. The architecture consisted of five dense layers, with the first layer comprising 128 neurons and the final output layer having 7 neurons to match the number of target classes. Activation functions like ReLU and sigmoid were used for hidden layers, while the output layer used a softmax function for multi-class classification. To prevent overfitting, dropout regularization was applied to the first hidden layer.

The model was compiled with the Adam optimizer and sparse categorical cross-entropy as the loss function, with accuracy and F1 score as evaluation metrics. Over 100 epochs, the model achieved a training accuracy of **83.01%** and a weighted F1 score of **27.22%**, while the validation accuracy and F1 score were **55.76%** and **0.2705%,** respectively. Despite reasonable training performance, the gap in validation metrics suggests potential overfitting or challenges in generalizing to unseen data.

3. K Nearest Neighbours

In this analysis, a K-Nearest Neighbors (KNN) classifier was employed with k=5 neighbours to predict the target variable. The model's performance was evaluated using 5-fold cross-validation, yielding accuracy scores across the folds ranging from **83.7%** to **85.3%**, reflecting its consistency. Following cross-validation, the KNN model was fitted to the entire training dataset and tested on the holdout test set. The model achieved an accuracy of **84.15%** and a weighted F1 score of **84.05%**.

4. Support Vector Classifier

Support Vector Classifier (SVC) with a polynomial kernel was utilized to model the target variable. The model's performance was assessed using 5-fold cross-validation, resulting in accuracy scores ranging from **72.6%** to **74.3%,** indicating consistent performance across folds. After training the SVC model on the full training dataset, predictions were made on the test set. The model achieved a test accuracy of **74.18%** and a weighted F1 score of **0.7389** which was poor compared to some of the other classifiers.

5. XG Boost

In this analysis, an XGBoost Classifier was employed to optimize predictions through a systematic hyperparameter tuning process using GridSearchCV. The grid search explored combinations of key hyperparameters, including learning rate, lambda, and alpha, across specified ranges. After a thorough 5-fold cross-validation, the best hyperparameters were identified as **learning_rate=0.2, lambda=2, alpha=0.1.**

The optimized XGBoost model, trained with these parameters, was evaluated on the test set, achieving an impressive accuracy of **90.32%** and a weighted F1 score of **0.9034.** These results demonstrate the model's ability to accurately predict the target variable while maintaining balanced performance across all classes, underscoring the effectiveness of hyperparameter tuning in boosting model performance.

Results and Evaluation

Scoring metrics

- **Accuracy**: The primary evaluation metric.

- **Leaderboard Position**: My submission scored **90.8**, closely trailing the highest score of **91.1**.

Model Performance Summary

The model performance scores are summarised in the table below

Model Name	Accuracy	F1 Score
Random Forest	0.895231	0.895202
K-Nearest Neighbors	0.841522	0.840473
Support Vector Machine	0.741811	0.738876
XG Boost	0.903179	0.903376

Conclusion

This project exemplifies my ability to:

1. Analyze and preprocess complex datasets efficiently.
2. Apply advanced machine learning techniques to achieve near-optimal results.
3. Use ensemble methods and hyperparameter tuning to maximize predictive accuracy.

Key Takeaways

- Achieving a score of **90.8** in a competitive environment highlights my problem-solving skills and technical proficiency.
- The process reinforced my understanding of the importance of feature engineering and model tuning.

Appendix

https://www.kaggle.com/competitions/playground-series-s4e2/overview