Integrating Body Signals to Uncover Patterns in Drinking Habits: A Novel Dataset Approach

Fariha Tasnim Chowdhury
Ahsanullah University of Science and Technology
Department Of CSE

Zakia Sultana
Ahsanullah University of Science and Technology
Department Of CSE

Humana Khan
Ahsanullah University of Science and Technology
Department Of CSE

Sheikh Aminul Islam
Ahsanullah University of Science and Technology
Department Of CSE

***Abstract*—Consuming alcohol has a lot of detrimental consequences on a person's health and day-to-day functioning.People even endanger their lives due to the harmful habit of regularly taking alcohol.In this paper, we deployed seven machine learning algorithms, particularly Logistic Regression,SVM,K-nearest neighbour,Naive Bayes,XG Boost,Random Forest,ADA Boost to a dataset of 22 input variables and one output variable. A number of models were compared with each other. For the evaluation of the models, we used five metrics : Accuracy,F1-score,Recall,Precision and Area under the curve. The results showed that, of all the scenarios we tested—apply on raw data, after feature extraction, and after feature extraction with specific features ,after feature extraction using selected feature obtained from feature selection—random forest produced the best accuracy of 0.7264 percentage in the first two scenarios and logistic regression produced the best accuracy of 0.7261 percentage in the final scenario.**

keywords: Machine learning, feature selection, feature extraction, alcohol consumption

I. Introduction

Alcohol consumption is one of the most prevalent and harmful behaviors in the world, affecting millions of people and causing various diseases and disorders, such as liver cirrhosis, cardiovascular disease, cancer, and mental illness. According to the World Health Organization (WHO), more than 2 billion people drink alcohol, and about 3.3 million people die each year from alcohol-related causes. Therefore, it is essential to develop effective methods for detecting and preventing excessive alcohol consumption among individuals and populations. It is a topic of interest in various fields such as healthcare, sociology, and psychology. Understanding drinking habits can provide valuable insights into an individual's health and social behavior. However, determining whether a person consumes alcohol can be a challenging task due to the myriad of factors involved. This paper presents a comprehensive study on the classification of individuals based on their alcohol consumption habits. Utilizing a dataset of approximately 100,000 instances with binary target variables indicating 'Yes' or 'No' for alcohol consumption, we aim to predict whether a person drinks alcohol or not. Various machine learning algorithms are employed to build predictive models, and their performance is evaluated and compared. The objective of this study is not only to achieve high accuracy in classification but also to gain insights that may be beneficial for health and social studies. In this paper, we tackle this problem using a data-driven approach. We have at our disposal a dataset comprising nearly 100,000 instances, each representing an individual. The target variable is binary, indicating whether the person is an alcohol drinker ('Yes') or not ('No'). Our approach involves applying various machine learning algorithms to this dataset to classify individuals into drinkers and non-drinkers. The algorithms we explore range from traditional methods like Logistic Regression and Decision Trees to more complex ones like Neural Networks. The rest of the paper is organized as follows. Section II reviews the related work. Section III describes the dataset and the data preprocessing steps. Section IV presents the machine learning algorithms and the evaluation metrics. Section V reports and discusses the experimental results. Section VI concludes the paper and suggests some future work.

II. Related Work

Kirstin Aschbacher [1] collected data from a commercially available smart breathalyzer. The collection included 33,452 different users' 973,264 unique BrAC observations. He modeled the data and predicted BrAC values using ensemble tree techniques. Additionally, he examined how well the ML system performed in comparison to users' subjective estimations of BrAC and discovered that the method performed 21 percent better in predicting when a user was legally intoxicated than the user's subjective estimate.

Adrienne Bergh [2] processed text message data from The BlackBerry Project, an investigation of the digital interactions of teenagers, using the doc2vec algorithm. When the author compared doc2vec to conventional manual coding, she discovered benefits in the program's efficiency and unsupervised nature. As a result, text data transformation became more effective and unsupervised, especially when evaluating informal and fragmented vocabulary.

Sangwon Bae [3] identified instances of alcohol consumption among young individuals by using supervised machine learning and sensor data from smartphones. Over the course of

28 days, 38 volunteers completed daily surveys and supplied sensor data, allowing them to gather data. The study classified episodes of not drinking, low-risk drinking, and high-risk drinking by comparing various machine learning models and sensor properties. The top-performing model classified drinking episodes with a high degree of accuracy by using Random Forest with 30-minute windows and three days' worth of historical data.

Anne H. Berman [4] conducted a review of seven randomized outcome studies on mobile interventions targeting hazardous drinking among university students. The research encompassed interventions like automated phone systems, text messaging, and smartphone applications. The effectiveness of these interventions in lowering risky drinking behaviors was compared by the authors. The outcomes differed; some programs indicated a decrease in the risk of drinking throughout the intervention period, whereas other interventions did not differ significantly from control groups. The authors stressed the necessity for additional study and the creation of quick, easy interventions that follow tried-and-true methods in order to address risky alcohol use among college students.

III. Dataset and Data preprocessing

The dataset employed for our analysis originates from Kaggle and is sourced from the National Health Insurance Service in Korea. This comprehensive dataset consists of a substantial 991,346 rows, each characterized by 23 columns. Among these columns, 22 serve as input variables, while one column is dedicated to the output variable, namely "class". The output variable classifies individuals into two categories: "drinker" and "not drinker." Notably, the dataset exhibits a balanced class distribution, with an equal representation of both classes; specifically, 50 percent of the instances are labeled as "drinker," and the remaining 50 percent as "non-drinker." This balanced distribution ensures that the dataset encompasses a representative sample of both classes, contributing to the robustness and reliability of our analysis and findings.

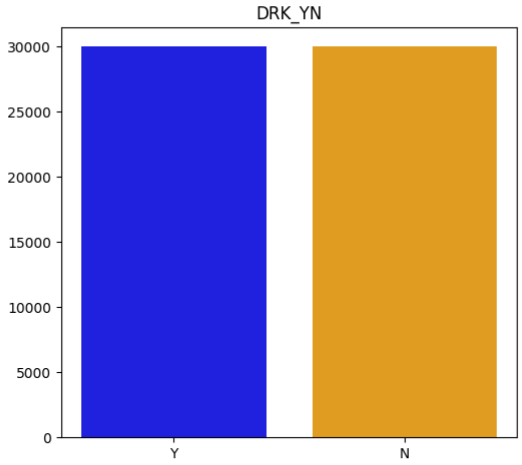

Fig. 1. Data distribution of 'Drinker' and 'Nondrinker' classes.

We perform some data preprocessing steps to clean and prepare the data. The data preprocessing steps are as follows: 1)We take 30000 data from each class and make 60000 data and shuffle the data due to excess runtime . 2) We convert the data of object type into data of numeric type using label encoder. 3) We split the dataset into training and testing sets, using 80 percent of the data for training and 20 percent of the data for testing.

IV. Methodology

A. Experimental Setup

We perform some feature selection method to choose the most relevant features that contribute the most to the output variable or result.Feature selection is a critical preprocessing step in machine learning, aimed at enhancing model performance and interpretability. The objective is to pinpoint and retain a subset of features that exert the greatest impact on the output variable, thereby streamlining the model and reducing complexity.

In our approach, we have utilized both wrapper methods, such as Forward Selection and Backward Elimination, as well as filter methods, including ANOVA, Chi-square, and Variance Threshold. Wrapper methods assess subsets of features by training and evaluating the model iteratively, while filter methods independently evaluate each feature based on statistical metrics or variance.

ANOVA (Analysis of Variance): ANOVA helps find features with significant mean differences across groups, aiding in selecting features that contribute to group distinctions.

Chi-square: Chi-square assesses independence between categorical variables, making it useful for identifying features associated with the target variable, especially in categorical data.

Variance Threshold: Variance thresholding removes low-variance features, streamlining the model by discarding less informative features with minimal variability.

Forward Selection: Forward selection adds features iteratively based on individual performance, progressively enhancing the model by systematically including promising features.

Backward Elimination: Backward elimination removes features iteratively, starting with all and eliminating the least significant ones, efficiently refining the model based on selected criteria.

5)We perform 3 feature extraction method .Feature extraction is another method of dimensionality reduction. Instead of selecting features, it creates a new set of features by transforming or combining the original features. The new features, also known as components, can capture the essential information in the original data but in a lower-dimensional space. The methods used for feature extraction include:

Principal Component Analysis (PCA): PCA simplifies data by creating new features that capture the most important information. It does this by finding combinations of the original features that hold the most variability, helping to reduce complexity.[fig 2]

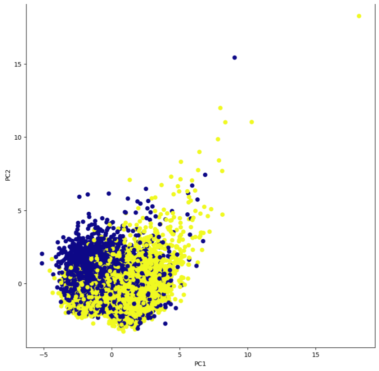

Fig. 2. After applying PCA

Linear Discriminant Analysis (LDA): LDA is like PCA but also considers class differences. It transforms features to maximize both variance and the distinction between classes, making it useful for classification problems.[fig 3]

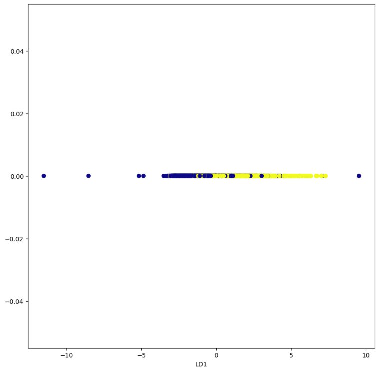

Fig. 3. After applying LDA

t-Distributed Stochastic Neighbor Embedding (t-SNE): t-SNE visualizes high-dimensional data in a simpler way. It groups similar data points together in a lower-dimensional space, making it great for exploring and understanding patterns in complex datasets.

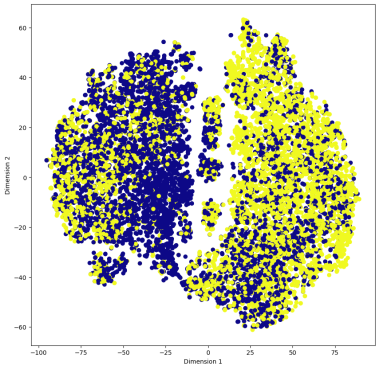

Fig. 4. After applying t-sne

After the data preprocessing steps, we obtain a dataset with 60000 rows and 23 columns, of which 22 are input variables and one is the output variable (class).

B. Machine Learning Model

We have selected a diverse set of seven machine learning models to comprehensively evaluate the dataset. These models include Naive Bayes, Random Forest, Logistic Regression, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), AdaBoost, and XGBoost.

Each model offers unique strengths and characteristics, ranging from the simplicity of Naive Bayes to the robustness of ensemble methods like Random Forest and XGBoost. Naive Bayes is a probabilistic classifier based on Bayes' theorem, Random Forest is an ensemble learning method that constructs multiple decision trees, Logistic Regression is a linear model used for binary classification, KNN is a non-parametric method that classifies data based on similarity to its nearest neighbors, SVM is a powerful algorithm for both classification and regression tasks by finding the optimal hyperplane, AdaBoost is an ensemble method that combines multiple weak classifiers to create a strong classifier, and XG-Boost is an efficient and scalable implementation of gradient boosting algorithms.

We apply machine learning model in four scenario . In the first scenario, we apply a machine learning model without any feature selection or extraction, using the entire set of features as input.

For the second scenario, we enhance the model by incorporating feature selection techniques. We systematically evaluate different feature selection methods to identify the one that yields the best results in terms of model performance.

In the third scenario, we take the selected features from the feature selection process and further improve the model through feature extraction. This involves transforming or combining the chosen features to create a more refined set, and we assess the model's performance using various machine learning algorithms.

In the fourth senario we applied machine learning model after performing feature extraction on the entire set of features .and after performing feature extraction we apply several classification model .

These scenarios help us understand the impact of feature selection and extraction on the overall effectiveness of the machine learning model.

V. Result Analysis

A. Before Feature Selection

In this scenario, we used the raw data with 22 input variables to train and test the 7 models. The performance metrics for each model are shown in Table1 From tab:Table1, We can see that the Random Forest model is the top performer with an accuracy of 0.7264. It also excels in other metrics, having the highest F1-score and recall among all models. This suggests that it's not only good at making correct predictions, but also at identifying the positive class.

The Logistic Regression and AdaBoost models are also competitive, with accuracies around 0.7245 and 0.72.40 and

TABLE I
PERFORMANCE METRICS OF THE MODELS ON THE RAW DATA WITHOUT FEATURE SELECTION OR EXTRACTION

Model	Accuracy	F1-score	Precision	Recall	AUC
Logistic Regression	0.7245	0.7240	0.7249	0.7231	0.80
SVM	0.7185	0.7172	0.72	0.7145	0.79
KNN	0.6908	0.6911	0.6898	0.6925	0.76
Naive Bayes	0.6871	0.6648	0.7151	0.6211	0.74
XGBoost	0.7202	0.7206	0.7190	0.7222	0.80
Random forest	0.7264	0.7288	0.7218	0.7359	0.80
AdaBoost	0.7240	0.7233	0.7247	0.7219	0.81

TABLE II
PERFORMANCE METRICS OF THE MODELS ON THE DATA WITH FEATURE SELECTION

Model	Accuracy	F1-score	Precision	Recall	AUC
Logistic Regression	0.7235	0.7222	0.7251	0.7192	0.80
SVM	0.7194	0.7189	0.7195	0.7182	0.80
KNN	0.6956	0.6990	0.6908	0.7074	0.77
Naive Bayes	0.6594	0.5850	0.7475	0.4805	0.76
XGBoost	0.7206	0.7215	0.7187	0.7242	0.80
Random forest	0.7241	0.7252	0.7218	0.7286	0.80
AdaBoost	0.7246	0.7247	0.7239	0.7256	0.81

AUC values of 0.80 and 0.81. Logistic regression's other performance metrics better than AdaBoost.

The XGBoost model has an accuracy of 0.7202, meaning it correctly predicted the outcome about 72% of the time. The F1-score and precision are both around 0.72, indicating a good balance between precision (correctly identified positives) and recall (true positives). The AUC (Area Under the Curve) score is 0.80, which is quite good - it means the model is capable of distinguishing between the positive and negative classes

The SVM model is slightly behind with an accuracy of 0.7185 and an AUC of 0.79. It's still a good model, but not as strong as the others in this comparison.

The KNN and Naive Bayes models are at the bottom of the pack. The KNN model has an accuracy of 0.6908 and an AUC of 0.76, while the Naive Bayes model has an accuracy of 0.6871 and an AUC of 0.74. The Naive Bayes model, in particular, struggles with recall, meaning it's not as good at identifying the positive class compared to the other models.

B. After Feature Selection

In this scenario we take the most relevant feature by applying feature selection on the dataset to predict the classification.To improve our model's performance, we experiment with different combinations of features. After trying various methods, we find that the best results come from forward feature selection.

Forward feature selection involves starting with no features and adding them one by one, selecting the one that improves the model's performance the most each time.

After conducting this process, we find that the optimal feature set for our model consists of 14 features along with 1 target variable. These 14 features are the ones that, when combined, provide the most accurate predictions for our classification task. The performance metrics for each model are shown in tab:Table2.

Here we can see that, The AdaBoost model is the top performer with an accuracy of 0.7246. It also excels in other metrics, having the highest AUC value of 0.81. This suggests that it's not only good at making correct predictions, but also at identifying the positive class.

The Random Forest model is another strong performer, with an accuracy of 0.7241 and a high F1-score of 0.7252. This suggests a good balance between precision and recall, meaning it's good at both identifying the positive class and avoiding false positives.

The Logistic Regression, SVM, and XGBoost models are also competitive, with accuracies around 0.72 and AUC values of 0.80. They have similar performance metrics, indicating that they are quite reliable in their predictions.

The KNN model is slightly behind with an accuracy of 0.6956 and an AUC of 0.77. It has a higher recall than precision, meaning it's better at identifying the positive class but may also have more false positives.

The Naive Bayes model had the lowest performance metrics among all. Especially, it had a lower recall value of 0.4805, which means it didn't do a great job in identifying the positive class.

In conclusion, while all these models have their strengths and weaknesses, the AdaBoost and Random Forest models stand out as the top performers in this analysis.

.

C. After Feature Extraction

In this scenario we perform feature extraction on all the 22 features .Feature extraction is used to reduce the dimensionality of the dataset by transforming the original features into a lower-dimensional space while preserving the most important information. After experimenting

TABLE III
PERFORMANCE METRICS OF THE MODELS ON THE DATA WITH FEATURE EXTRACTION

Model	Accuracy	F1-score	Precision	Recall	AUC
Logistic Regression	0.7245	0.7240	0.7249	0.7231	0.80
SVM	0.7185	0.7172	0.72	0.7145	0.79
KNN	0.6908	0.6911	0.6898	0.6925	0.76
Naive Bayes	0.6871	0.6648	0.7151	0.6211	0.74
XGBoost	0.7202	0.7206	0.7190	0.7222	0.80
Random forest	0.7264	0.7288	0.7218	0.7359	0.80
AdaBoost	0.7240	0.7233	0.7247	0.7219	0.81

We can see among the models evaluated, Random Forest emerges as the top performer, boasting an accuracy of 0.7264 and the highest F1-score of 0.7288, indicative of its balanced precision and recall. With a recall of 0.7359, it excels in correctly identifying positive cases. AdaBoost follows closely with an accuracy of 0.7240 and the highest AUC of 0.81, showcasing its ability to distinguish between classes effectively.

Logistic Regression and XGBoost exhibit robust performances, each with accuracies around 0.72 and AUC values of

0.80, indicating reliable predictive capabilities. SVM, slightly trailing behind, still demonstrates commendable performance with an accuracy of 0.7185 and an AUC of 0.79.

KNN, while slightly lower in accuracy at 0.6908, displays a notable recall of 0.6925, suggesting proficiency in identifying positive cases. However, it may also incur more false positives.

On the contrary, Naive Bayes lags behind with the lowest performance metrics, including a notably low recall value of 0.6211, indicating challenges in correctly identifying positive instances. Here we can see that compared to feature selection in feature extraction accuracy of Logistic Regression,XGBoost,Random Forest,Naive Bayes accuracy slightly better . After feature extraction we can say that it performed well.

D. After Feature Extraction with Selected Features

In this scenario, we conduct feature extraction on the chosen feature set comprising 14 features and 1 target variable, which were obtained through feature selection.

TABLE IV
PERFORMANCE METRICS OF THE MODELS ON THE DATA WITH FEATURE SELECTION AND FEATURE EXTRACTION WITH PCA

Model	Accuracy	F1-score	Precision	Recall	AUC
Logistic Regression	0.7261	0.7258	0.725	0.726	0.80
SVM	0.7194	0.7189	0.7195	0.7182	0.80
KNN	0.699	0.7017	0.6938	0.7098	0.77
Naive Bayes	0.6607	0.5853	0.7499	0.4799	0.75
XGBoost	0.7197	0.7205	0.7168	0.7243	0.80
Random forest	0.7194	0.7185	0.7190	0.7181	0.80
AdaBoost	0.7251	0.7245	0.7244	0.7246	0.80

Here we can observe some notable differences when feature selection and feature extraction with PCA (Table IV) are applied.

The Logistic Regression model shows a slight improvement in Table IV with an accuracy of 0.7261, F1-score of 0.7258, and recall of 0.726, compared to its performance in Tables I, II, and III.

The SVM model shows consistent performance across all four tables. In Table IV, where feature selection and feature extraction with PCA are applied, the SVM model has an accuracy of 0.7194, F1-score of 0.7189, precision of 0.7195, and recall of 0.7182. These metrics are similar to those in the other tables. Therefore, the application of feature selection and feature extraction with PCA does not significantly improve the performance of the SVM model.

The K-Nearest Neighbors (KNN) model also exhibits a significant enhancement here. The accuracy increases to 0.699, the F1-score to 0.7017, and the recall to 0.7098, which are all higher than the corresponding values in the other tables.

For the Naive Bayes model, although there is a slight increase in accuracy (0.6607) and F1-score (0.5853) in Table IV compared to Table I table II, these values are still lower than those in Tables III.

The performance of the XGBoost remains relatively consistent across all tables, with no significant improvement observed in Table IV.

The performance of the Random Forest did not able to imporve here . The accuracy of random forest 0.7194. which is lower than the previous scenario.

Lastly, the AdaBoost model in Table IV shows a slight improvement in all performance metrics (accuracy of 0.7251, F1-score of 0.7245, precision of 0.7244, and recall of 0.7246) compared to its performance in the other tables.

In conclusion, while the AdaBoost, Logistic Regression, and KNN models show better performance in Table IV compared to the other tables, the improvements are relatively minor.

VI. CONCLUSION

In this paper, we used a dataset of 60,000 data with 22 input variables and one binary output variable to use seven machine learning algorithms to evaluate alcohol consumption.We experiment the models' performance using feature extraction and feature selection seperately and feature extraction and feature selection together and raw data without feature extraction or selection.Accuracy, F1-score, precision, recall, and AUC were the five metrics we used to assess the models. Four scenarios were used by us.Among them,the random forest model performs the best on the raw data and after using feature extraction, according to our results. After using feature extraction, the accuracy did not change.Logistic regression achieved the best accuracy after using feature extraction using selected features.

REFERENCES

[1] Aschbacher, K., Hendershot, Aschbacher, K., Hendershot, C.S., Tison, G. et al. Machine learning prediction of blood alcohol concentration: a digital signature of smart-breathalyzer behavior. npj Digit. Med. 4, 74 (2021). https://doi.org/10.1038/s41746-021-00441-4

[2] Bae, S., Chung, T., Ferreira, D., Dey, A. K., Suffoletto, B. (2018). Mobile phone sensors and supervised machine learning to identify alcohol use events in young adults: Implications for just-in-time adaptive interventions. Addictive Behaviors, 83, 42-47. Ambulatory Assessment of Addictive Disorders. https://doi.org/10.1016/j.addbeh.2017.11.039.

[3] Berman, A.H., Gajecki, M., Sinadinovic, K. et al. Mobile Interventions Targeting Risky Drinking Among University Students: A Review. Curr Addict Rep 3, 166–174 (2016). https://doi.org/10.1007/s40429-016-0099-6

[4] Kim SY, Park T, Kim K, Oh J, Park Y, Kim DJ. A Deep Learning Algorithm to Predict Hazardous Drinkers and the Severity of Alcohol-Related Problems Using K-NHANES. Front Psychiatry. 2021 Jul 9;12:684406. doi: 10.3389/fpsyt.2021.684406. PMID: 34305681; PMCID: PMC8299053.

[5] A. Bergh, "A machine learning approach to predicting alcohol consumption in adolescents from historical text messaging data," M. S. thesis, Chapman University, Orange, CA, 2019. https://doi.org/10.36837/chapman.000072