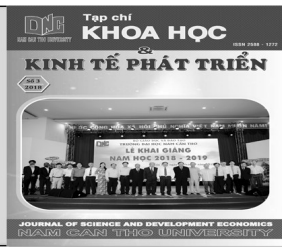
Journal of Science and Development Economics
Nam Can Tho University

Website: jsde.nctu.edu.vn



Diagnosing the quality of wine using an adapting decision tree classifier for streaming data

Vo Ngoc Truong Duy[1], Vo Van Phuc[1], Tran Duy Khang[2], Ngo Ho Anh Khoi[1]

[1]*Faculty of Information Technology, Nam Can Tho University*
[2]*Faculty of Engineering and Technology, Nam Can Tho University*

Received: November 10, 2023
Revised: November 30, 2023
Accepted: December 5, 2023

Keywords: *AI application, decision tree algorithm, wine quality diagnosis*

Từ khóa: *chuẩn đoán, chất lượng rượu vang, thuật toán cây quyết định, ứng dụng AI*

ABSTRACT

The research is focused on exploring the applications of Artificial Intelligence algorithms in handling diagnostic wine quality data. The article discusses the successful implementation of the Decision Tree algorithm for this purpose. This drives the main research goal, which revolves around integrating the Decision Tree with flexible sliding window techniques that can continuously adapt and update over time. The primary objective of the study is to address the wine quality diagnostic problem. Alongside this goal, there are additional smaller objectives to achieve. The initial step involves studying and researching theoretical foundations and measurement methods, as well as analyzing wine quality. Lastly, the goal of deploying a test application is set, aiming to create a Wine Quality Diagnostic Page. The interface of the page is designed to be user-friendly, intuitive, and informative about the functioning and content of the wine quality diagnostic method.

TÓM TẮT

Nghiên cứu tập trung vào việc khám phá các ứng dụng của thuật toán Trí tuệ nhân tạo trong việc xử lý dữ liệu chẩn đoán chất lượng rượu. Bài viết thảo luận về việc triển khai thành công thuật toán Cây quyết định cho mục đích này. Điều này thúc đẩy mục tiêu nghiên cứu chính xoay quanh việc tích hợp Cây quyết định với các kỹ thuật cửa sổ trượt linh hoạt có thể liên tục thích ứng và cập nhật theo thời gian. Mục tiêu chính của nghiên cứu là giải quyết vấn đề chẩn đoán chất lượng rượu vang. Bên cạnh mục tiêu này, còn có những mục tiêu nhỏ hơn cần đạt được. Bước đầu tiên bao gồm việc tìm hiểu, nghiên cứu cơ sở lý thuyết và phương pháp

đo lường cũng như phân tích chất lượng rượu. Cuối cùng, mục tiêu triển khai ứng dụng thử nghiệm được đặt ra nhằm tạo Trang Chẩn đoán Chất lượng Rượu. Giao diện của trang được thiết kế thân thiện với người dùng, trực quan và cung cấp nhiều thông tin về chức năng cũng như nội dung của phương pháp chẩn đoán chất lượng rượu.

1. INTRODUCTION

The issue at hand is understanding the significance of wine and why it is produced in various regions worldwide, including Vietnam. The state of wine in Vietnam: Despite the challenges posed by tropical viticulture, it is evident that quality wine can be produced in Vietnam. Vietnamese wine is crafted from the Cardinal grape variety, classified as a table grape in France. Additionally, there are a few Vitis Vinifera varieties - Cabernet Sauvignon, Chardonnay, Syrah - but they are relatively scarce. The importance of wine in Vietnam: Ladora Winery in Phat Chi – Da Lat, Lam Dong, has exclusively invested in a 6-hectare winery and a 20-hectare vineyard in Ninh Thuan to create a line of high-quality wines under the Vietnamese brand. The government and the Ministry of Tourism have also shown interest in investing more in this industry, as wine tourism is on the rise. Therefore, the significance of wine in Vietnam lies not only in its economic potential but also in its contributions to the domestic tourism sector.

According to various articles and international research reports, Sangodkar et al (2021) [1] explored the application of machine learning models for predicting wine quality; similarly, (Bhardwaj et al, 2022) [2], Piyush Bhardwaj introduces RF and AdaBoost models as machine learning classifiers to predict wine quality. The author evaluates these models based on accuracy, precision, recall, and is developing a web application based on machine learning for researchers and wine growers to predict wine quality using chemical and physical compounds present in their wines. Another study, by K. R. Dahal, author employs the Wine quality dataset from UCL to demonstrate the feasibility of using various statistical analyses to predict wine quality based on different parameters. This study implies that wine quality can be forecasted even before production, suggesting an alternative approach for understanding the variables influencing wine quality. In the Vietnamese context, (Hoàng Anh Lê, 2004) by Hoàng Anh Lê, the author focused on the significance of wine production. Moreover, in (Bùi Công Danh and Nguyễn Thị Diệu Hiền, 2021) [3] authors aims to analyze the importance of wine and the research timeline related to sensory evaluation, improvement, and new product creation. The dataset used in this research is named "Winequality_white.csv."All these works collectively share the primary objective of investigating wine quality prediction through the application of machine learning techniques. Decision Tree Analysis of Wine Quality Data was updated by Raj Parmar in 2019 (Parmar, 2019) [4]. The article focuses on utilizing the Decision Tree algorithm in a study related to wine quality diagnosis. The algorithm is implemented through a series of steps,

including library importation, dataset loading, data splitting into features and targets, division into training and testing sets, model construction, training on the training set, making predictions on the test set, evaluating model accuracy, and making predictions for new data. The model achieved a relatively high accuracy on the test set, indicating its effectiveness as a classifier. However, accuracy could be improved through adjusting model hyperparameters or employing different machine learning algorithms. Overall, the decision tree classifier is a powerful and versatile algorithm suitable for various classification tasks. The underlying datasets were downloaded from Kaggle. Assessing Wine Quality Using a Decision Tree was last updated by Seunghan Lee, and his colleagues in September 2015 (Lee et al., 2015) [5]. This article also focuses on the Decision Tree algorithm in the context of evaluating wine quality using decision trees. Wine quality assessment is crucial for the wine industry, and accurate evaluations are important for producers, distributors, and consumers. In recent years, decision trees have increasingly been used to predict wine quality ratings. This article summarizes Seunghan Lee's research on enhancing wine quality ratings using decision trees. The research is divided into three parts: research overview, research methodology, and significance and limitations of the study. The objective of Seunghan Lee's research is to improve wine quality ratings through the application of decision trees. Decision trees are a machine learning technique that employs a tree model to predict the value of a target variable based on multiple input variables. In this study, decision trees are utilized to predict wine quality ratings based on physical characteristics such as acidity, pH, alcohol concentration, and residual sugar.

The research is focused on exploring the applications of Artificial Intelligence algorithms in handling diagnostic wine quality data. The article discusses the successful implementation of the Decision Tree algorithm for this purpose. This drives the main research goal, which revolves around integrating the Decision Tree with flexible sliding window techniques that can continuously adapt and update over time. The primary objective of the study is to address the wine quality diagnostic problem. Alongside this goal, there are additional smaller objectives to achieve. The initial step involves studying and researching theoretical foundations and measurement methods, as well as analyzing wine quality. Lastly, the goal of deploying a test application is set, aiming to create a Wine Quality Diagnostic Page. The interface of the page is designed to be user-friendly, intuitive, and informative about the functioning and content of the wine quality diagnostic method.

2. MATERIALS AND METHODS

During the data exploration phase for the research topic, a multitude of datasets were discovered. However, three datasets exhibited the most comprehensive attributes and highest availability: Wine_Quality_Data by Ghassen Khaled (Khaled, 2023) [6], last updated on April 14, 2023, comprises 13 columns and 6497 rows of data. The data fields include various parameters like fixed acidity, volatile acidity, citric acid, residual sugar, chloride, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol content, quality, and label. This dataset is employed for training models

that classify white or red wines. Red Wine Quality from UCI Machine Learning, last updated in 2018, consists of 12 columns and 1599 rows of data. The data attributes encompass characteristics such as fixed acidity, volatile acidity, citric acid, residual sugar, chloride, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol content, and quality. This dataset is utilized to discern whether a wine is red or not. Wine Quality Data Set (Red & White Wine) by Ruthgn (2021) [7], updated in 2021, comprises 13 columns and 6497 rows of data. The attributes of this dataset are akin to the Wine_Quality_Data by Ghassen Khaled, differing primarily in parameter values and label positions. All of these datasets were found on Kaggle.

The enumerated datasets are all relevant to wine prediction, but only one dataset meets the requirements of the research topic. The other datasets present multiple issues that render them unsuitable for use in this study. For a dataset to be applicable, it must consist of numerical data, have specific class labels, and feature numerous attributes to yield objective outcomes. Among the listed datasets, only the Wine_Quality_Data fulfills these criteria. It provides the necessary capabilities for the research topic, making it the chosen dataset for this internship project.

This dataset may be temporary, as various new wine types have emerged, leading to potential changes in dataset classification in the future due to increased diversity. Hence, it can be identified as a dataset in an unstable environment. Consequently, the use of data classification for conventional machine learning algorithms within a static environment is not feasible. Old methods do not support classification and require advanced algorithms capable of processing data in dynamic environments. Currently, existing databases face the challenge of static concentration over time due to being trained using classical algorithms (in practice). This phenomenon occurs only once; when new data arrives, the previously learned information must be retrained entirely. For example, if data set 1 is used to create a model and new data set 2 arrives, data set 1 must be retrained from scratch alongside data set 2 to build a new model). Moreover, in the context of modern reality, where data environments evolve over time, training must occur continuously in real-time, and model predictions must be regularly updated. Consequently, data learning must transpire within an evolving data environment, which means that testing methods will continuously learn in a non-static environment. Several methods have been applied to transform classical algorithms into continuous learning approaches, replacing them with sliding window methods to evolve traditional machine learning techniques into advanced ones. The description of the Sliding Window-based approach is as follows:

Taking into account the evolution of concepts in an evolving data environment, the most recent training data is determined within a defined time window (either based on a time interval or several data instances). This approach can involve reclassifying "groups" (within the data selected by the temporary window) or updating the model if the online learning method permits. In this case, the process of "forgetting" (as mentioned above) is automatically managed by this learning method. This type of approach typically involves three steps: Detecting concept changes using

statistical tests across different windows. If an observed change occurs, select representative and recent data to adjust the models. Updating the models. The window size is predetermined by the user. The key point of these methods lies in determining the window size. Most methods employ a fixed-size window configured for each real-world problem. This way, classical algorithms can be applied in dynamic environments, but they lack the characteristics of progressive machine learning (they don't reuse stored data, only the model is used for improvement). Therefore, the historical part of the following algorithms focuses on presenting incremental machine learning algorithms, which have been researched and developed in recent years.

After transforming the data from its raw form to standardized format for use in the software, the dataset must be stored in a file with the ".csv" extension. Following all data normalization processes, we will obtain a standardized dataset consisting of 13 features, totaling 6497 data points, which encompass various parameters. Raw data must undergo preprocessing before being applied to the training process of the software. From these raw data, labels are transformed where the label "red," representing the color of red wine, is converted to the number "1," while the label "white," indicating the color of the second type of wine, is converted to "0." The remaining attributes used for assessment, such as fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulfates, alcohol, and quality, remain unchanged as they are already in numerical form and do not require conversion. These indices have been sourced from the aforementioned data and have been verified by domain experts. Each feature will have the following indices:

Fixed acidity: is a measure of non-volatile or non-volatile acids. These acids are derived from grapes. The main fixed acids found in wine are tartaric, malic, citric, and succinic acids. The unit of measurement is (g/L).

Volatile acidity: is a measure of the easily vaporizable (or gaseous) acids in wine. The primary volatile acid in wine is acetic acid, which is also the main acid associated with the aroma and taste of vinegar. The unit of measurement is (g/L).

Citric acid: is a weak organic acid, added to wine as a natural preservative and to enhance the acidity of the wine.

Residual sugar: is the amount of natural sugars left over from the grape after the fermentation process in red wine concludes. It is measured in (g/L).

Chlorides: In wine, the presence of 2 to 4 (g/L) of mineral acids, different from other organic acids, contributes to the potential salty taste of the wine. Thus, chlorides play a crucial role in the wine's saltiness. The unit of measurement is (g/L).

Free sulfur dioxide: is a measure of the amount of SO_2 that is not bound to other molecules and is used to calculate molecular SO_2. It is used during winemaking to prevent oxidation and the growth of microorganisms. The unit of measurement is (mg/L).

Total sulfur dioxide: is a measure of the combined and free forms of SO_2. Bound SO_2 refers to SO_2 molecules bound to other compounds, mainly aldehydes, pyruvates, and anthocyanins. It is used in winemaking to prevent oxidation and the growth of

microorganisms. Excessive levels of SO_2 can inhibit the fermentation process and result in undesirable sensory effects. The unit of measurement is (mg/L).

Density: Unit of measurement (g/cm^3).

pH: Describes the acidity or alkalinity of the wine on a scale from 0 (very acidic) to 14 (very alkaline); most wines fall within the pH range of 3-4.

Sulphates: Chemical formula $(SO_4)^{2-}$. It is an additive in wine that can contribute to the production of sulfur dioxide (SO_2) gas, acting as an antimicrobial and antioxidant agent. The unit of measurement is (g/L).

Alcohol: Percentage of alcohol content in the wine.

Quality: Rating scale from 1 to 10 based on sensory data, ranging from 3 to 8 in this dataset.

Predicted Label: This parameter carries a decision value and is particularly important. This label has only two possible values, either "0" or "1". If the wine is red, the label will be 1, while if the wine is white, the label will be 0. The prediction includes a total of 1599 data points for the red type and 4898 data points labeled as white.

The dataset used in this experiment consists of two parts: the training data portion includes 4545 data points (constituting 70% of the original data), and the testing data portion contains 1952 data points (representing 30% of the original data). The positions of these data points will vary in each experiment, and for each experiment, they will be randomly shuffled, both before training and after training.

The experiment will be conducted using an asynchronous batch learning model (Batch Learning) with a batch size of 4545. This means that the system will execute 4,545 steps, with approximately 129 data points per step. The model utilized in the experimental phase is a batch-wise data grouping model. This model will use the same dataset, organized into batches, by dividing the original dataset into smaller groups using 35 steps. Each batch of data contains 129 data points. This quantity strikes a balance between being not too large and not too small, facilitating the experimentation process.

All achieved results are based on Balanced Accuracy. Balanced Accuracy is a metric that can be used to evaluate the performance of a binary classifier. It is particularly useful when classes are imbalanced, meaning one of the two classes appears much more frequently than the other. Using Balanced Accuracy is significantly more complex than using regular Accuracy. Regular Accuracy simply computes the percentage ratio of a dataset's subgroup based on the total available data. Initially, this works well, but when the data is highly skewed (e.g. one data point in class A, 999 data points in class B), the accuracy calculation loses its correctness. To address this issue, a calculation method was devised, relying on true negative and true positive values, allowing for the computation of true negatives, true positives, false negatives, and false positives percentages. With all these parameters at hand, the formula for Balanced Accuracy can be employed to calculate the most accurate and optimal percentage truthfully. The Balanced Accuracy formula used in these experiments is as follows.

First, you need to refer to the confusion matrix:

	True Ground Truth Labels (1)	Incorrect Ground Truth Labels (0)
True positive predictions (1)	TP	FP
False positive predictions (0)	FN	TN

Figure 1. Confusion Matrix

In the matrix above, "positive" or "negative" in TP/FP/TN/FN refer to the predictions made, not the actual labels. (Thus, "false positive" is the case of incorrectly predicting positive). Below are the formulas for sensitivity and specificity based on the confusion matrix:

Sensitivity Formula: Sensitivity = TP/(TP + FN)

Specificity Formula: Specificity = TN/(TN + FP)

Balanced Accuracy Formula: $\text{Balanced accuracy} = \frac{(\text{Sensitivity} + \text{Specificity})}{2}$

The dataset used in the experiment consists of two parts: one for training and the other for testing. The training dataset comprises 1400 rows, while the testing dataset contains 701 rows. This dataset will vary in each experiment, but the data itself remains unchanged – only shuffling is applied to maintain data integrity. The model employed in the experiment follows a batch data approach. This model uses the same dataset in batches, achieved by partitioning the original dataset into smaller groups over 1400 steps. Each batch contains one data point, as this batch size is considered manageable and efficient in achieving optimal results for the task.

3. RESULTS AND DISCUSSION

By applying artificial intelligence techniques, specifically algorithms like Decision Trees, in combination with the Sliding Window method, a more equitable approach to assessing the authenticity of data can be achieved. Utilizing charts to compare the average outcomes of the three algorithms, this method ensures a fairer representation of data accuracy during the comparison of results from various algorithms. The experimental average results of the algorithms are depicted in the chart below (Figure 2).

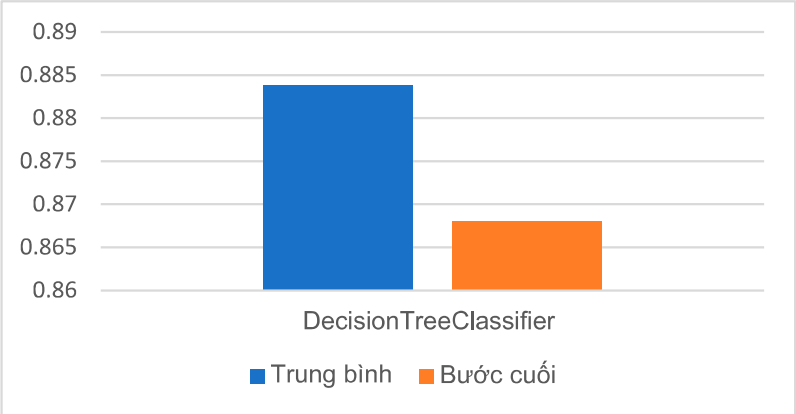


Figure 2. Chart of Average Percentage of Experimental Algorithm by Age (Decision Tree)

Looking at the data on the chart, we can analyze the average ratios and the final step's quite favorable ratio of the Decision Tree algorithm as follows:

The performance of the Decision Tree algorithm is remarkably stable, consistently achieving above 85% accuracy. The data illustrates that the Decision Tree algorithm maintains a consistent and noteworthy performance, with an average accuracy of over 85% (specifically 88.38%), and accuracy in the final stage also exceeding 85% (with an accuracy of 86.39%). This is a significant advantage, demonstrating the Decision Tree's capability to provide accurate predictions in the majority of cases, making it applicable to the wine quality diagnosis problem.

In addition to calculating the algorithm's average results, an alternative approach such as analyzing the experimental model results based on age reveals a more comprehensive, detailed perspective. This allows for a visual assessment that aids in arriving at the most accurate conclusions. The experimental model results based on age are depicted in the chart below (Figure 3).

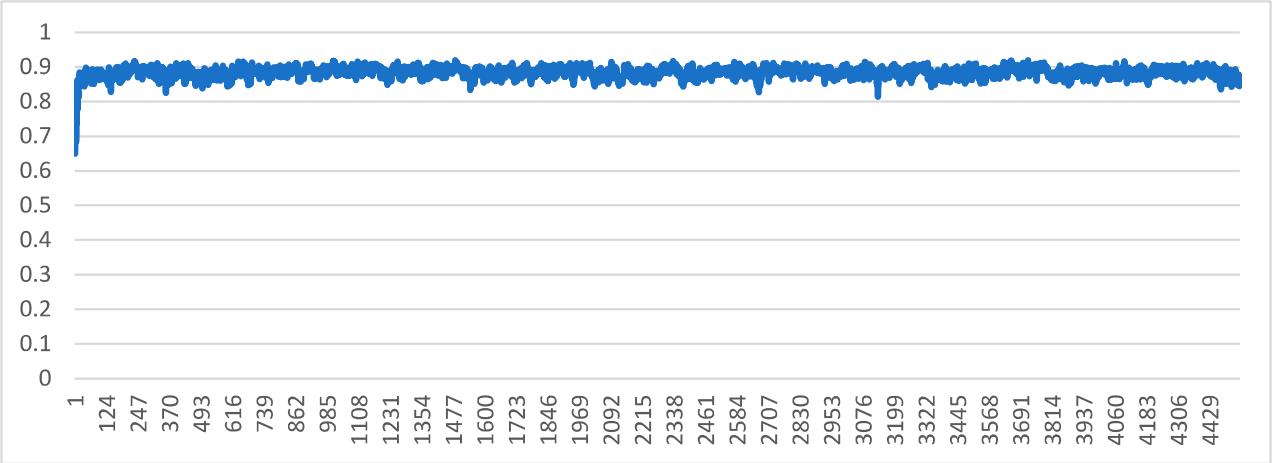


Figure 3. Chart of Experimental Model Results by Age (Decision Tree)

Examining the chart above, we observe that the Decision Tree algorithm starts with a relatively low point of 64% and gradually increases during the first 10 to 30 steps, after which it stabilizes. Looking at the graph, it's evident that the Decision Tree algorithm demonstrates reasonable stability. However, the highest accuracy rate of the Decision Tree can reach up to 91.72%, which is a notable figure when compared against various other algorithms. Specifically, during the phase from step 998 to 1008, as shown in the chart below (Figure 4).

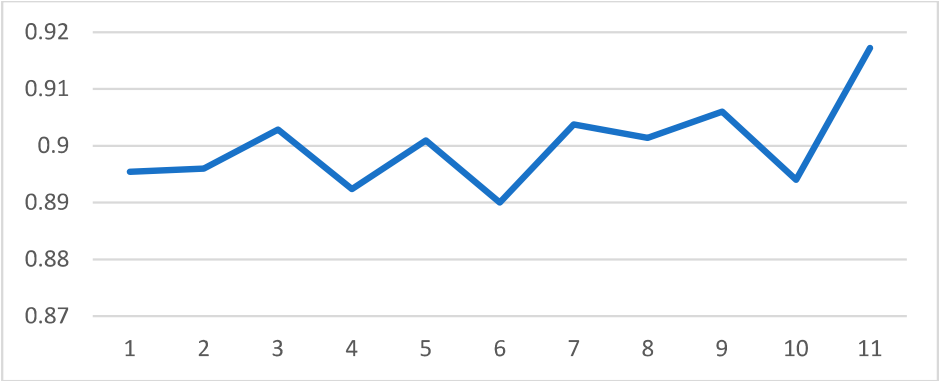


Figure 4. Peak Phase of the Decision Tree Algorithm (Batch 998-1008)

Upon examining the chart, it is evident that the Decision Tree consistently achieves a gradual increase, always surpassing the 80% mark. This is a rare occurrence when compared to other algorithms, as the Decision Tree demonstrates its ability to handle non-continuous and missing value data effectively. This capability minimizes the need for extensive data preprocessing and allows the algorithm to perform efficiently across various types of data. Despite encountering phases where results may not be optimal, the Decision Tree consistently stabilizes quickly, as depicted in the chart below (Figure 5).

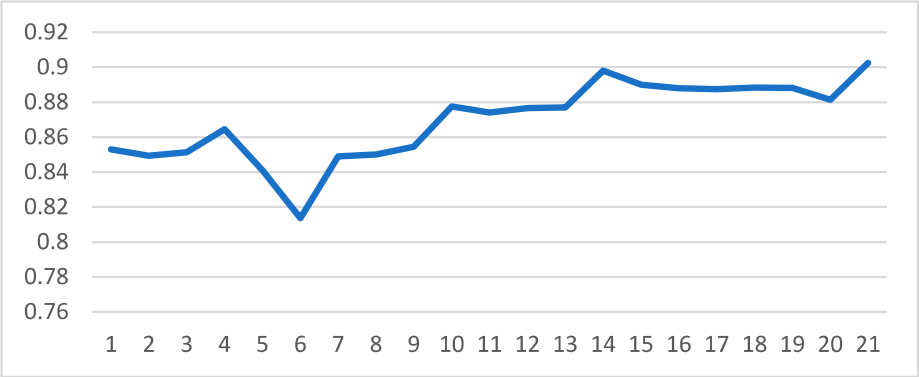


Figure 5. Lowest Dropped Phase of the Decision Tree (Batch 800 - 820)

At step 3135, the lowest accuracy rate achieved by the Decision Tree was 81.37%. However, by step 3139, the algorithm quickly rebounded to its stable performance with an accuracy of 87.74% and continued to gradually increase in subsequent steps. The accuracy rates during this phase do not vary significantly, ranging from 81.37% to 87.74%. The variation in performance is relatively small, indicating a consistent level of stability in the model.

In conclusion, the Decision Tree algorithm demonstrates numerous advantages and consistent performance with accuracy rates exceeding 85%. This underscores that the Decision Tree can be a solid and useful choice for various prediction and classification tasks. However, it's essential to note that each algorithm has its own strengths and weaknesses. The selection of an appropriate algorithm also depends on the specific requirements of the problem and the characteristics of the data.

Implementation of a real world application:

Based on the final results highlighted in the previous section, the chosen algorithm for addressing the problem is the Decision Tree algorithm. The project will encompass various functional nodes, including prediction functionality, execution of classical algorithms, a list of processed models, system configuration, and user authentication. This application will be implemented within a website environment, organized into two main user roles: Algorithm Setup (admin or developer) and Diagnosis User (end user). These roles are depicted in the use case diagram below: [Use Case Diagram illustrating the roles and functionalities of the application]**.** In this system, the Algorithm Setup role involves functions related to configuring and managing algorithms, while the Diagnosis User role focuses on utilizing the prediction capabilities and accessing processed models. The implementation will enable efficient interaction and utility for both types of users within the website framework.

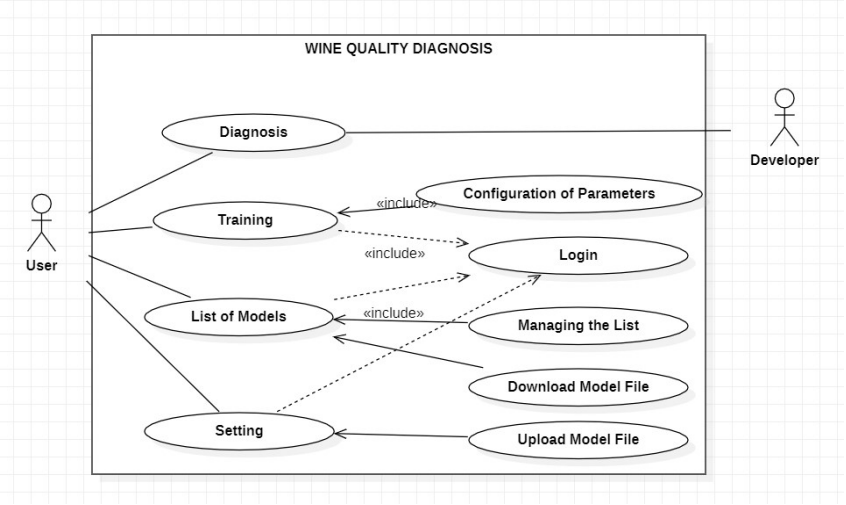


Figure 6. Use-Case Diagram of the System

To install the program, you need to first download the installation file named "StressDiagnostic.rar". After extracting the file, you will find a folder named "StressDiagnostic". To run the software, the user's computer needs to have certain Python libraries and Python 3.9.9 installed. Upon extraction, you will see a folder named "SETUP". Inside this folder, you will find a file named "python-3.9.9-amd64.exe", used to install Python 3.9.9, and a file named "inLib.bat" to install the necessary libraries required to run the software. Once the environment setup process is complete, a file named "Remove.bat" will be available. This file is used to delete unnecessary data files, including those used for testing purposes. It should be used in two scenarios: right after extraction and installation, and to remove all data from previous runs. To run the program, use the "Runserver.bat" file. This file is pre-configured to execute the command "py manage.py runserver", and the program will run on the default port "http://127.0.0.1:8000/". It's important to note that the user's computer should be connected to the internet at all times,

and the minimum system requirements include Windows 10, 2GB of RAM, and a 10GB or larger hard drive to ensure smooth and stable performance. After a successful installation, to use the software, access the port "http://127.0.0.1:8000" to reach the main page of the system. On the main interface page, a set of input fields will appear for data entry to perform predictions. Below are examples of the forms built within the "Wine Quality Diagnosis" system.

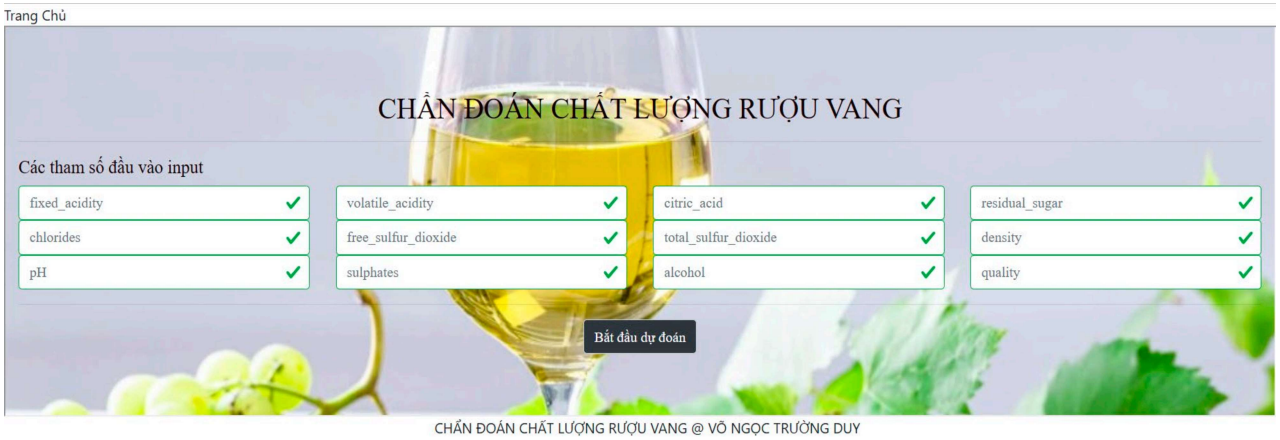


Figure 7. Main Interface of the Wine Quality Diagnosis System

5. CONCLUSION

Upon completing the research and report composition process, a comprehensive evaluation of the results can be presented. The report's content is clear and elaborates on data, charts, and algorithms specifically. The theoretical contribution is significant, as this project developed an algorithm with various training options. Simultaneously, the system was developed as a website interface, a rare feat achieved by very few systems. It tackled the challenge of dynamic data fluctuations by evolving the algorithm towards model-based training, whereas other algorithms usually only cater to static data through data-based training. However, this project only reached the research stage, leaving room for many future expansions. These may include implementing automated raw data processing within the system, optimizing model training processes, enhancing the system's interface for smoother user experience, refining the code for better aesthetics and broader user accessibility, and bringing the application into practical use, facilitating quick wine quality diagnoses for users. The research and development of this project were conducted in the context of the Vietnamese market, where there's a prevalence of counterfeit alcohol impacting both quality and human health. The system was built upon three Decision Tree algorithms as they fulfilled the requirements for changing data, as mentioned above.

REFERENCES

[1] Sangodkar, V.P. (2021). *Wine Quality Prediction Using Machine Learning.* https://www.ijraset.com/fileserve.php?FID=37629.

[2] Bhardwaj, P. (2022). *A machine learning application in wine quality prediction.* https://www.sciencedirect.com/science/article/pii/S266682702200007X.

[3] Bùi Công Danh, Nguyễn Thị Diệu Hiền (2021). *Đánh giá cảm quan rượu vang trắng bằng nổn nhân tạo.* https://sti.vista.gov.vn/tw/Lists/TaiLieuKHCN/Attachments/316303/CVv146S42021272.pdf.

[4] Parmar, R. (2019). *Decision Tree Analysis of Wine Quality Data,* https://www.kaggle.com/code/rajyellow46/decision-tree-analysis-of-wine-quality-data/notebook.

[5] Lee, S., Kang, K., & Park, J. (2015). *Assessing wine quality using a decision tree.* https://www.researchgate.net/publication/308862829_Assessing_wine_quality_using_a_decision_tree.

[6] Khaled, G. (2023). *Wine Quality Data*. https://www.kaggle.com/datasets/ghassenkhaled/wine-quality-data/discussion.

[7] Ruthgn (2021). *Wine Quality Data Set (Red & White Wine).* https://www.kaggle.com/datasets/ruthgn/wine-quality-data-set-red-white-wine?resource=download.