

## Project Name:- Wine Classifier App

**The Wine Classifier App is a graphical user interface (GUI) application built with Python that uses machine learning to predict the quality of wine based on its chemical properties (like acidity, alcohol , chloride, pH, etc.).**

## Submitted To:- Submitted By:-

Dr. Krishan Tuli Name:- Rekha Rani

UID:- 24MCI10261

Sec. :- 24MAM-3B

Github link :-

## ACKNOWLEDGEMENT

I would like to express my sincere gratitude to Krishan Tuli supported me in the successful completion of this project, **"Wine Classifier App using Machine Learning"**.

Firstly I would like to thanks Dr. Krishan Tuli for their continuous guidance, encouragement, and valuable feedback throughout the development of this project.

I also extend my appreciation to the creators and maintainers of open-source libraries such as **Pandas**, **NumPy**, **Scikit-learn**, **XGBoost**, and **Tkinter**, which played a key role in implementing the machine learning pipeline and building the graphical user interface.

Finally, I am grateful thanks to my friends and teacher’s for their constant motivation and moral support during the course of this project.

This project has been a great learning experience, and I am thankful for the opportunity to apply theoretical knowledge to a practical and real-world application.

## CERTIFICATE

This is to certify that project entitled “Wine Classifier App” has been successfully completed by Diksha.

The project is a Bonafide work and fulfils the requirements for the Masters of Computers Application and course code – 24CAP-672 conducted during my Second semester 2025 academic Session

The project demonstrates a comprehensive understanding of Python tkinter programming GUI development and client-server architecture. The student has successfully implemented a Wine Classifier App, showcasing their ability to apply theoretical knowledge to practical application.

**Supervisor Name**: Krishan Tuli

**Department**: University Institute of Computing **Institution name**: Chandigarh University

**Date:** 15/04/2025

**INDEX**

|  |  |
| --- | --- |
| **S. No.** | **Name** |
| **1** | **Acknowledgement** |
| **2** | **Certificate** |
| **3** | **Abstarct** |
| **4** | **Introduction** |
| **5** | **Objective** |
| **6** | **Methodology Used** |
| **7** | **Desgin** |
| **8** | **Source code** |
| **9** | **Output** |
| **10** | **Conclusion** |

# ABSTRACT

The Wine Classifier App is a machine learning-based desktop application designed to predict the quality of wine using its physicochemical properties. The primary objective of this project is to build a user-friendly system that assists users in determining whether a given wine sample is of good or bad quality.

The application is developed using Python and utilizes libraries such as Tkinter for the graphical user interface, Pandas and NumPy for data processing, and XGBoost for building an efficient and accurate classification model. The dataset used contains various chemical features of wine, including acidity, sugar content, pH, and alcohol levels.

During preprocessing, missing values are handled, features are scaled, and the data is split into training and testing sets. A binary classification approach is adopted where wines with a quality score greater than 5 are considered good, and others as bad. The trained model predicts the quality of wine based on user inputs provided through the GUI.

The application offers two modes of input: manual entry of wine features and search by Wine ID from the dataset. The model's prediction is displayed along with an informative message and color-coded feedback for better user understanding.

This project demonstrates how machine learning can be practically applied in the food and beverage industry to aid decision-making and ensure product quality.

# INTRODUCTION

Wine quality assessment has traditionally been a task handled by expert sommeliers, relying on sensory evaluation techniques such as taste, aroma, and appearance. However, with the advent of data science and machine learning, it is now possible to predict wine quality using measurable physicochemical properties. This project, titled **"**Wine Classifier App**"**, aims to implement a machine learning solution that can classify wine as either **g**ood or bad based on such features.

The application is built using Python, integrating powerful libraries such as Pandas, NumPy, Scikit**-**learn, and XGBoost for data handling and model training, along with Tkinter for creating an intuitive graphical user interface (GUI). The model uses a dataset containing various attributes of red wine samples, such as fixed acidity, volatile acidity, citric acid, residual sugar, pH, alcohol content, and more.

To simplify the classification, a binary label is used: wines with a quality score above 5 are labeled as good quality, and those with a score of 5 or below are labeled as bad quality. The machine learning model—XGBoost Classifier—is trained on preprocessed and normalized data to achieve accurate predictions.

The app enables users to either enter wine characteristics manually or search by Wine ID to autofill the fields. Once the data is input, the app predicts the wine's quality and provides visual feedback, making it easy for users to understand and interpret the results.

# OBJECTIVE

#  To collect and understand wine quality data:

# Utilize a structured dataset (WineQT.csv) containing real-world chemical properties and quality ratings of red wine samples.

# Analyze the dataset to identify key features that influence wine quality.

#  To process and prepare the data:

# Handle missing or inconsistent values using imputation techniques.

# Normalize or scale the data using MinMaxScaler to bring all features into a common range.

# Convert the multi-class quality labels into binary classification (Good vs Bad) for simplicity and clarity.

#  To build a robust prediction model:

# Implement the XGBoost Classifier, a high-performance ensemble learning algorithm, to train on the processed data.

# Split the dataset into training and testing subsets to evaluate model performance accurately.

#  To design an interactive GUI using Tkinter:

# Create input fields for users to enter wine attributes manually.

# Provide search functionality to autofill input fields using a specific Wine ID.

# Display helpful range hints next to each input field to guide user input.

#  To integrate the trained model into the GUI app:

# Accept real-time user inputs, transform them using the same preprocessing pipeline, and pass them to the model.

# Display prediction results as either “Good Quality Wine” or “Bad Quality Wine” along with relevant color-coded feedback and popups

#  To improve user experience and usability:

# Provide input validation and error handling to prevent crashes.

# Allow users to clear fields easily and re-enter new values.

# Ensure the GUI is visually clean, responsive, and intuitive.

#  To demonstrate a real-world use case of machine learning:

# Highlight how AI and ML can assist in automating quality assessment in the food and beverage industry.

# Provide an example of how data-driven tools can support quick, consistent, and objective decision-making.

#  To enable future enhancements:

# Structure the code to allow easy integration of new models or features (e.g., multi-class predictions, data visualization).

# Make the dataset and interface adaptable to other similar classification problems.

**Technologies used**:-

1) Anaconda

2) Jupyter Notebook

3) Tkinter.

4) GUI designing.

**Design Analysis:-**

* 1. **Data Loading and Validation**

The application starts by importing necessary libraries and loading the wine dataset using pandas. It checks for the existence of the file at the specified path and handles missing files with an appropriate error message. Additionally, it ensures that the dataset contains a unique identifier by adding an Id column if it doesn't already exist.

**2. Data Preprocessing**

Before training the model, the data undergoes thorough preprocessing. Any missing values are filled using the mean strategy through SimpleImputer. A new target column, best quality, is created by converting the quality scores into binary classes—where wines with quality above 5 are considered "good" (1) and the rest as "bad" (0). The dataset is then split into features and targets, and the features are scaled using MinMaxScaler to ensure all values are normalized between 0 and 1.

**3. Model Training**

The machine learning model used in this application is the XGBClassifier from the XGBoost library, known for its efficiency and high performance in classification tasks. The model is trained on the preprocessed training data with an evaluation metric of log loss. This trained model is then used to predict the quality of wine based on user inputs.

**4. Graphical User Interface (GUI) Design**

A visually appealing and user-friendly interface is designed using the tkinter library. The main window has a custom title, fixed dimensions, and a colored background to enhance visual aesthetics. Fonts and button styles are carefully selected for clarity and accessibility.

**5. Input Section Design**

The GUI dynamically generates input fields for all relevant wine features (excluding Id). Each field includes a label and a helpful range hint based on the dataset's minimum and maximum values. This guides the user to input values within the acceptable data range and helps prevent invalid entries.

**6. Search by Wine ID**

A unique feature of this application is the ability to search and auto-populate wine details using a specific Id. This functionality enables users to review or reuse previously entered data by simply entering the wine’s ID number. If a valid ID is found, all corresponding feature values are automatically filled in the input fields.

**7. Prediction Mechanism**

Once the user enters feature values, they can click on the "Predict Quality" button. The application processes the inputs, scales them accordingly, and uses the trained model to predict whether the wine is of good or bad quality. The prediction is displayed on-screen with color-coded text, and a pop-up message provides additional confirmation.

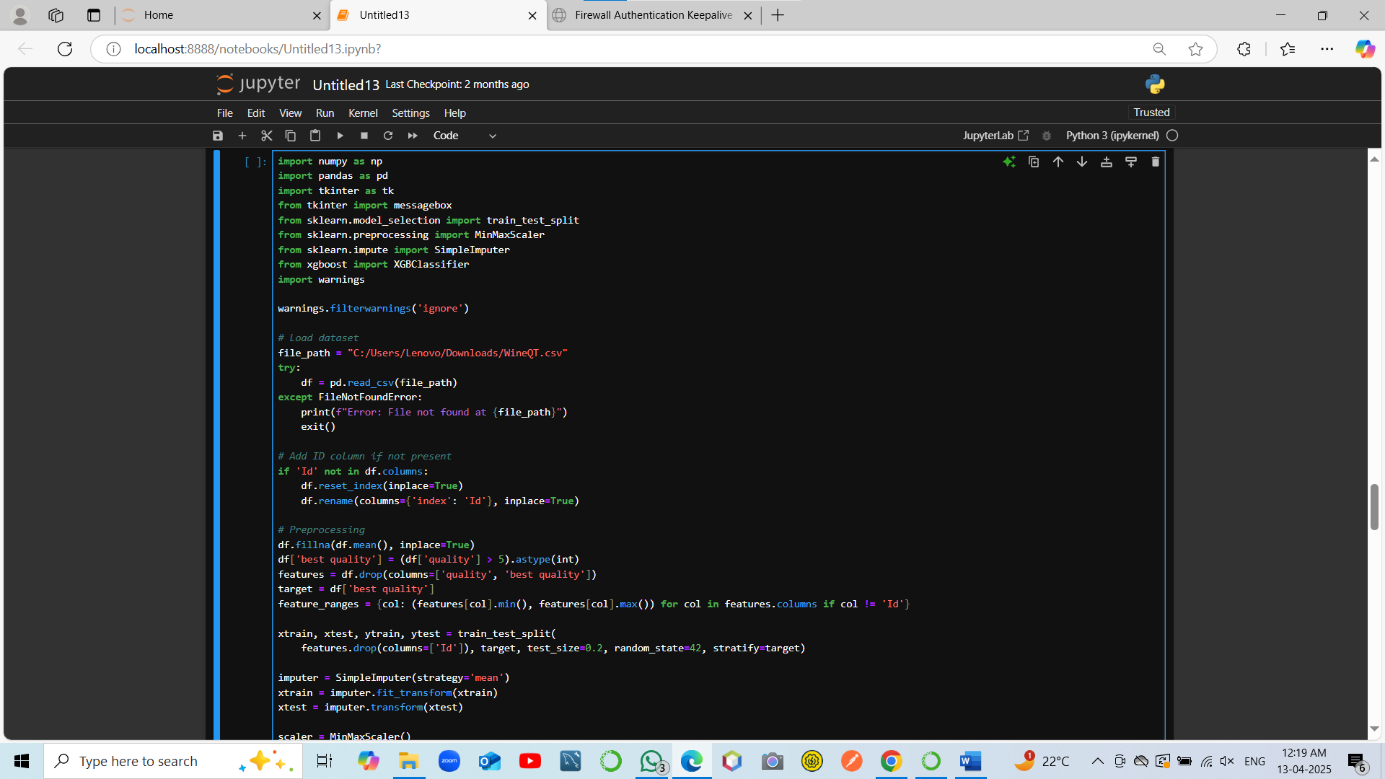
**8. Clear Functionality**

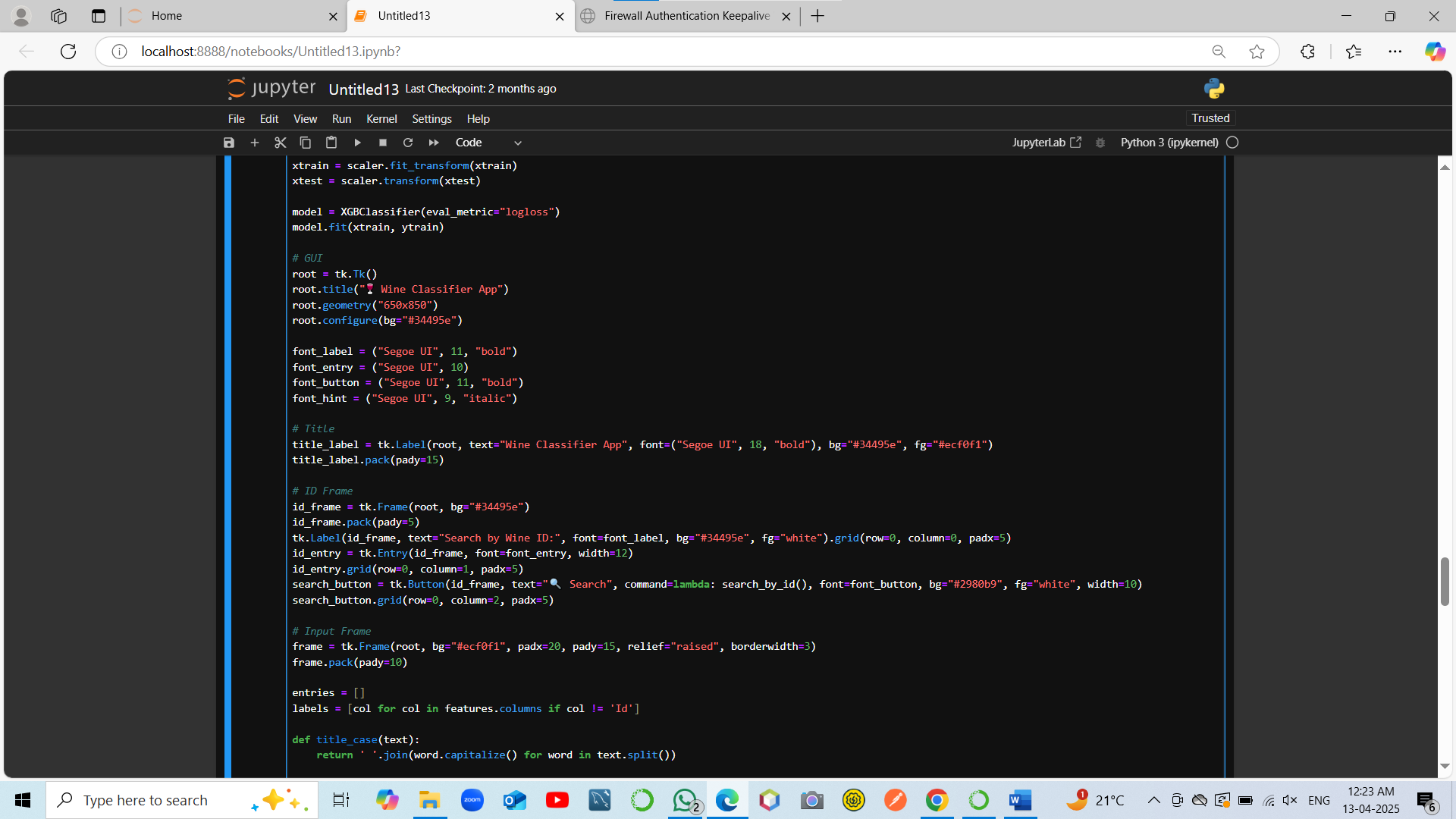
A "Clear" button is provided to reset all input fields and prediction results, making it convenient for users to perform multiple predictions without restarting the application. This also includes clearing the Wine ID field to prevent confusion.

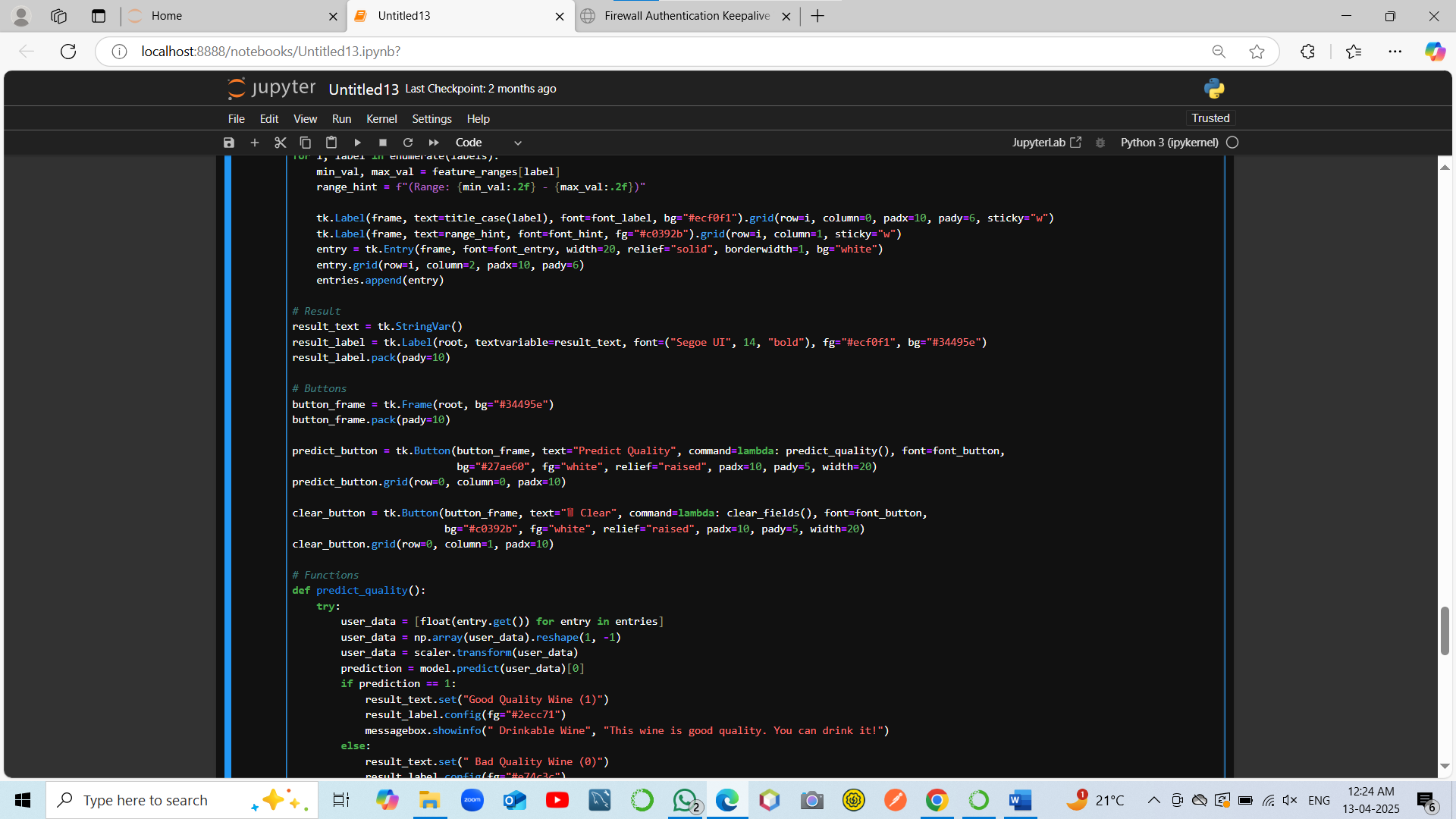
**9. Function-Based Modular Design**

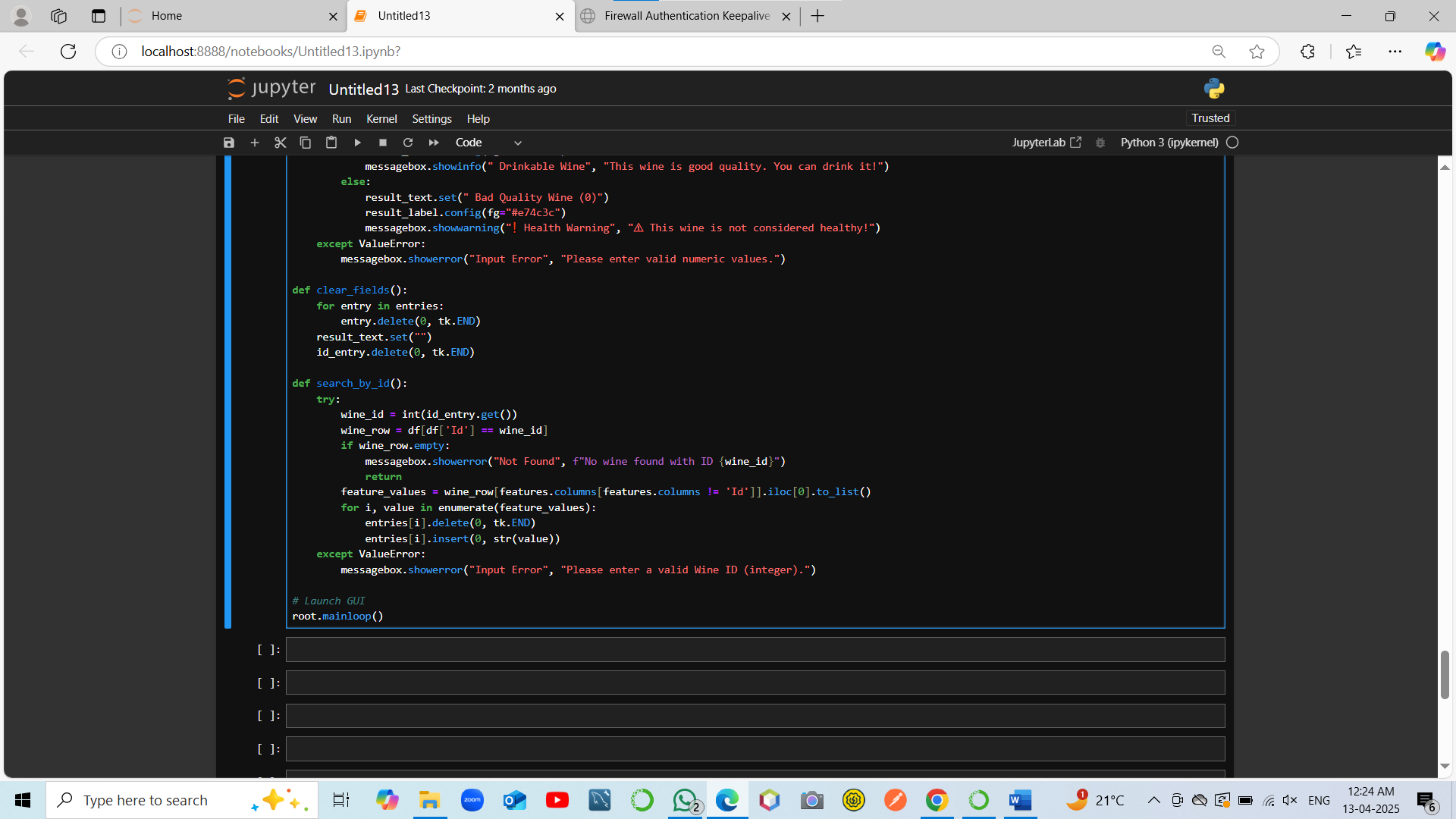
The program is structured using modular functions to improve readability and maintainability. Core functions like predict\_quality(), clear\_fields(), and search\_by\_id() handle specific tasks and keep the logic organized. This design allows for easier debugging, enhancements, and reuse of code.

**CODING:-**

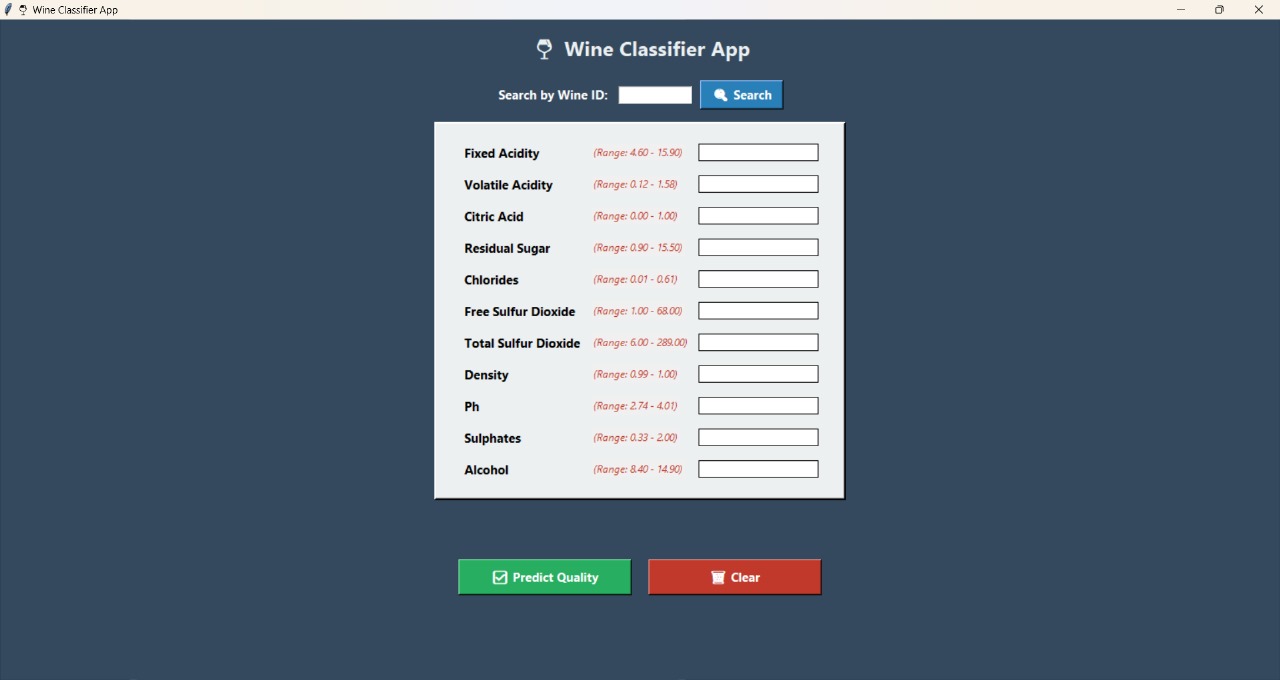
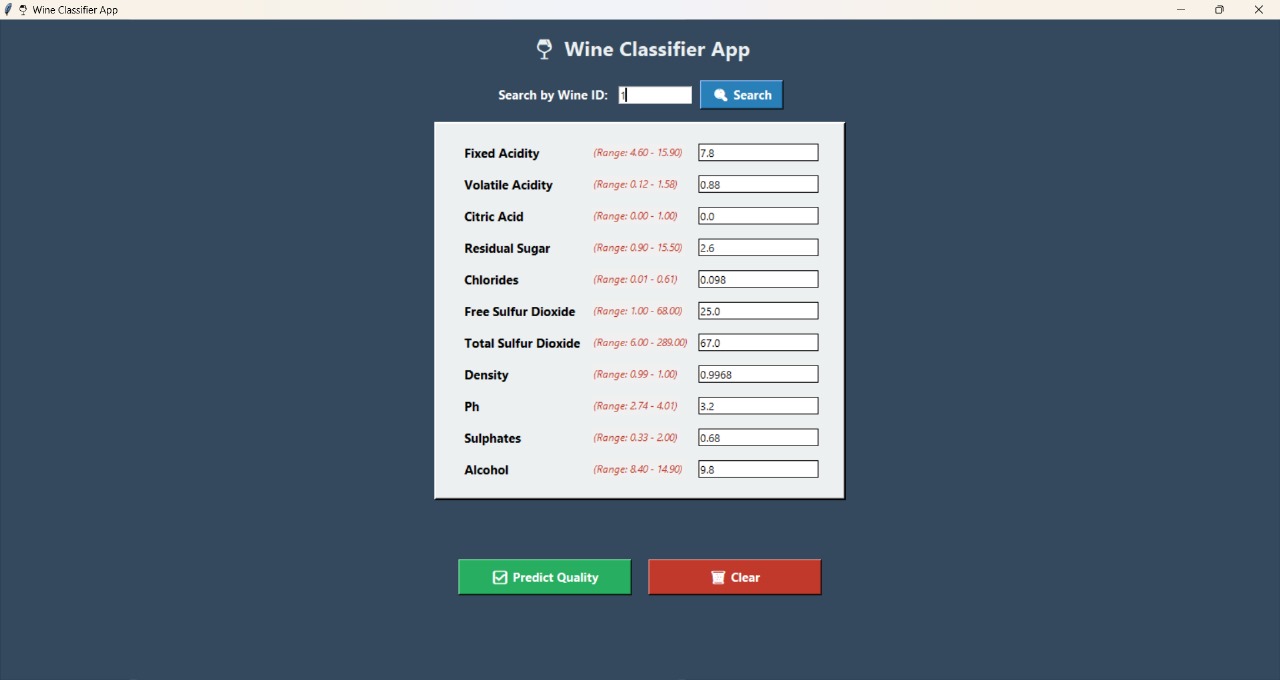
****

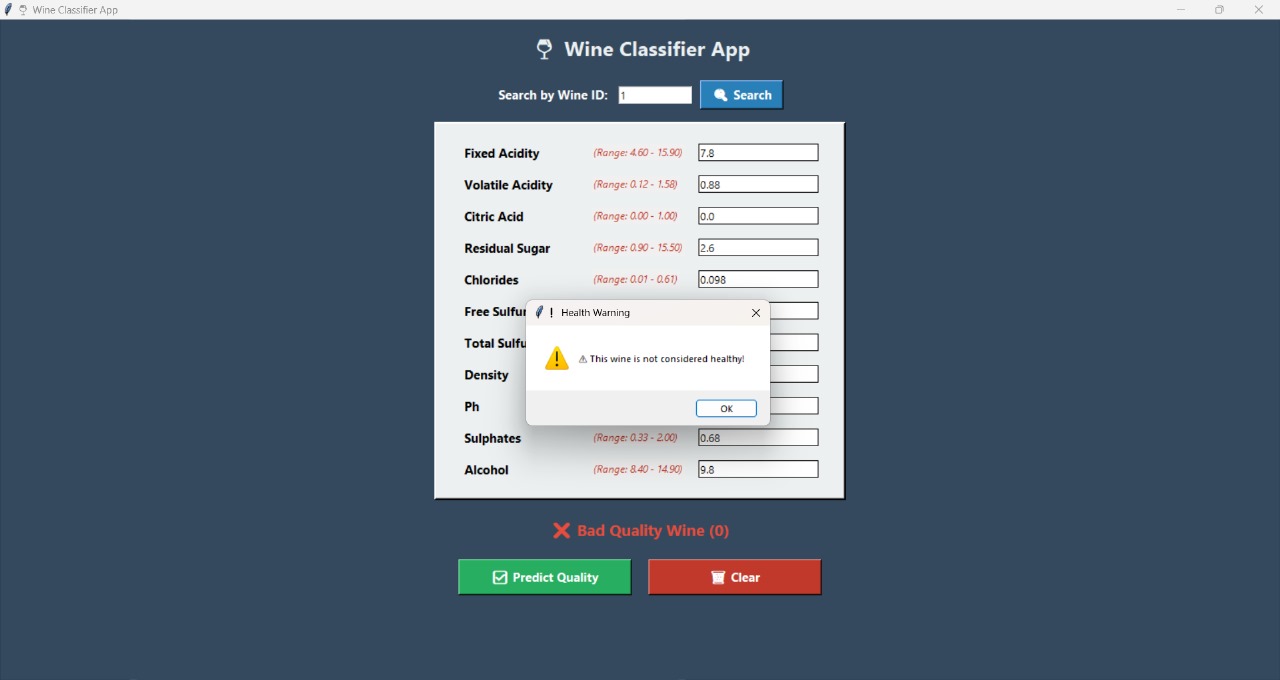
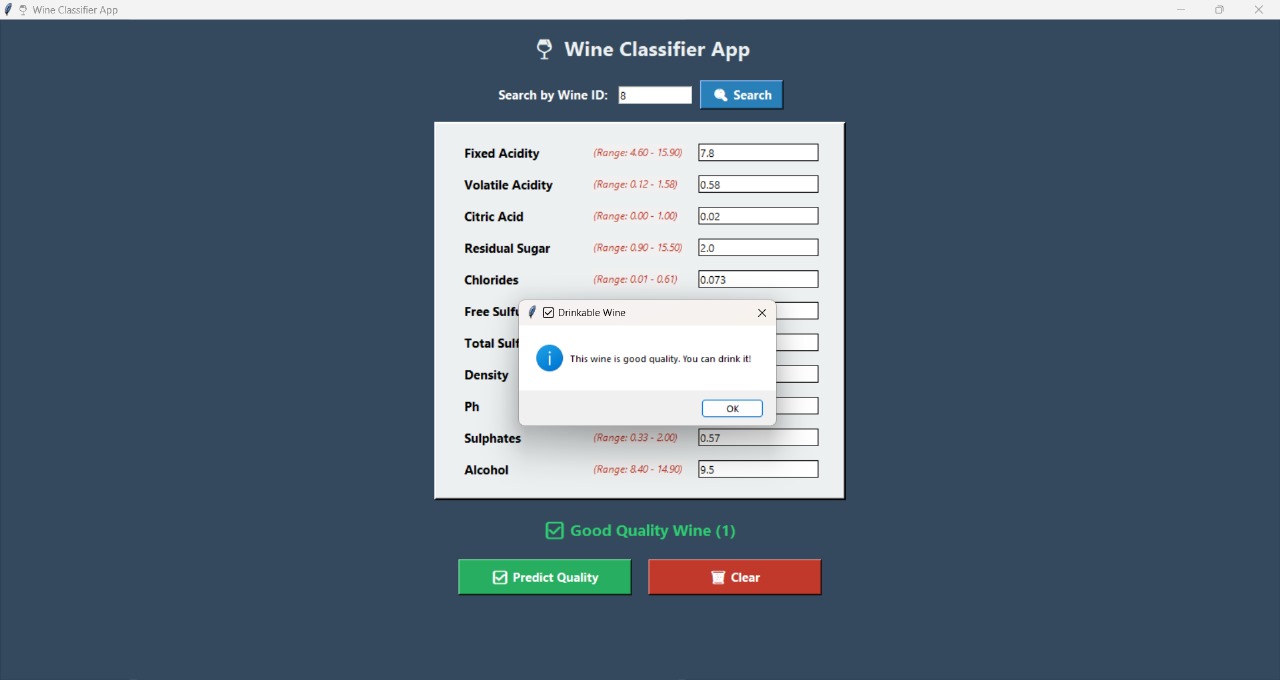
****





**OUTPUT:-**

** **

****  ****

**CONCLUSION:-**

* **Accurate Wine Classification**: The model effectively classifies wine quality as either good or bad using XGBoost, a powerful machine learning algorithm.
* **Smart Use of Data**: The app uses key wine chemical features and transforms them using scaling and imputation for better model performance.
* **User-Friendly GUI**: The Tkinter-based interface allows users to easily input data, search by wine ID, and get instant predictions—no technical background required.
* **Interactive Features**: Range hints, result color coding, and informative pop-ups enhance the user's interaction and understanding.
* **Real-World Application**: This project shows how machine learning can be integrated into real-time decision-making tools for the food and beverage industry.
* **Scalable and Extendable**: The system can be further extended to support multi-class predictions, include visual analytics, or integrate with a wine database/API.