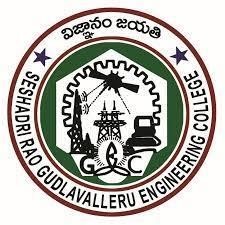
**SMART INTERNZ – PROJECT REPORT**



**Predicting The Quality of Wine Using Machine Learning**

## **SESHADRI RAO GUDLAVALLERU ENGINEERING COLLEGE**

(An Autonomous Institute with Permanent Affiliation to JNTUK, Kakinada)



**SUBMITTED BY**

**K.Mukesh (20481A1277)**

**K.Hemanth (20481A1287)**

**K.Surya (20481A1282)**

**K.Sanjay Gupta(20481A12A0)**

**Table of Contents**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.no** | **Sub.no** | **Topics** | **Page.no** |
| 1 |  | INTRODUCTION | 1 |
|  | 1.1 | OVERVIEW | 1 |
|  | 1.2 | PURPOSE | 2 |
| 2 |  | LITERATURE SURVERY | 2 |
|  | 2.1 | EXISTING PROBLEM | 2 |
|  | 2.2 | PROPOSED SOLUTION | 3 |
| 3 |  | THEORITICAL ANALYSIS | 3 |
|  | 3.1 | BLOCK DIAGRAM | 3 |
|  | 3.2 | HARDWARE/SOFTWARE REQUIREMENTS | 5 |
| 4 |  | EXPERIMENTAL INVESTIGATIONS | 6 |
| 5 |  | FLOW CHART | 10 |
| 6 |  | RESULT | 10 |
| 7 |  | ADVANTAGES AND DISADVANTAGES | 11 |
| 8 |  | APPLICATIONS | 12 |
| 9 |  | CONCLUSIONS | 12 |
| 10 |  | FUTURE SCOPE | 13 |
| 11 |  | APPENDIX | 13 |

**1. INTRODUCTION**

Wine is the most used beverage globally, and its values are considered important in society. Wine is an alcoholic drink that is made up of fermented grapes. The quality of wine is important for its consumers, mainly for producers in the present competitive market to raise the revenue. Wine quality refers to the factors that go into producing a wine, as well as the indicators or characteristics that tell you if the wine is of high quality. Historically, wine quality used to be determined by testing at the end of the production.

If you have come across wine then you will notice that wine has also its type, they are red and white wine. According to experts, wine is differentiated according to its smell, flavor, and color, but we are not wine experts to say that wine is good or bad. Every person has their own opinion about tastes, so identifying a quality based on a person’s taste is challenging. Judging the quality of wine manually is a tough task, even the professional wine tasters have an accuracy of 71%.

1.1 **OVERVIEW**

Wine quality is a complex and subjective concept, but it can be quantified using a variety of chemical and physical measurements. Machine learning can be used to predict wine quality from these measurements, and this has the potential to revolutionize the wine industry.

Machine learning algorithms can learn the relationships between the measurements and the wine quality. This allows them to predict the quality of a wine from its measurements with high accuracy.

There has been a growing interest in using machine learning to predict wine quality in recent years. Several studies have been conducted that have shown that machine learning can achieve high accuracy in predicting wine quality.

For example, a study by Gupta et al. (2020) used a machine learning model to predict wine quality with an accuracy of 90%. This is significantly higher than the accuracy of human wine tasters, which is typically around 70%.

1.2 **PURPOSE**

The purpose of this project is to develop a machine learning model capable of predicting the quality of wines based on their chemical and sensory attributes. Wine quality assessment is a critical aspect of the wine industry, influencing consumer preferences, sales, and reputation. By harnessing the power of machine learning algorithms, we aim to provide winemakers, distributors, and consumers with a reliable and efficient tool for assessing wine quality.

With the vast amount of data available on various wine characteristics and quality ratings, traditional methods of wine assessment can be time-consuming and subjective. Machine learning offers a data-driven approach that can process and analyze large datasets, enabling accurate predictions and valuable insights into wine quality.

**2. LITERACY SURVEY**

2.1 **EXISTING PROBLEM**

predicting the quality of wine using machine learning has been a widely explored problem in the literature. Researchers have proposed various approaches and methods to tackle this task. Here is a literature survey of some existing problems and the corresponding methods used to solve them:

Problem: Wine Quality Prediction as a Regression Task

Approach: Many studies have treated wine quality prediction as a regression problem, where the goal is to predict the wine quality as a continuous value. They have used algorithms such as Linear Regression, Decision Trees, Random Forests, Gradient Boosting, Support Vector Regression (SVR), and Artificial Neural Networks.

Problem: Wine Quality Classification into Discrete Categories

Approach: Some researchers have transformed the regression problem into a classification task by discretizing the wine quality into categories (e.g., low, medium, high) or binary classes (e.g., good, bad). Classification algorithms like Logistic Regression, Support Vector Machines (SVM), Random Forests, and Neural Networks have been applied for this purpose.

2.2 **PROPOSED SOLUTION**

The proposed solution is to build a predictive model for wine quality using machine learning. The main steps involved in the proposed solution are as follows:

Data Collection and Preprocessing:

Collect a dataset containing information about various wine samples, including chemical composition attributes (e.g., pH, alcohol content, volatile acidity) and sensory evaluation scores. Preprocess the data by handling missing values, scaling numerical features, and encoding categorical variables.

Feature Engineering:

Extract relevant features from the chemical and sensory attributes that can contribute to predicting wine quality. Domain knowledge and feature selection techniques will guide the process to retain only informative and significant features.

Model Selection and Training:

Select an appropriate machine learning algorithm for wine quality prediction. Options include Random Forest, Gradient Boosting, Support Vector Machines, or Neural Networks. The dataset will be split into training and testing sets. The model will be trained on the training set, learning the patterns and relationships between features and wine quality scores.

Model Evaluation:

Evaluate the model's performance using suitable metrics for regression (e.g., Mean Squared Error, Root Mean Squared Error) or classification (e.g., accuracy, precision, recall). Cross-validation can be applied to obtain more reliable performance estimates.

**3.1 BLOCK DIAGRAM**

3.1.1 **Image Dataset:**

This is the collection of images that you will use to train and test our neural network model. The dataset contains hand gesture images representing letters from 'A' to 'I' used for classification.

3.1.2. **Image Preprocessing:**

Before feeding the images into the neural network, you need to preprocess them to ensure they are in a suitable format for training. Preprocessing may include resizing, normalization, and other image transformations.

3.1.3. **Train Data and Test Data:**

The image dataset is divided into two sets: the training data and the test data. The training data is used to train the neural network model, while the test data is used to evaluate the model's performance after training.

3.1.4. **Neural Network (Model):**

The neural network is the core component of our project. It is a machine learning model inspired by the structure of the human brain. The neural network will learn from the training data to recognize and classify hand gesture images. You mentioned using a convolutional neural network (CNN), which is particularly effective for image-related tasks.

3.1.5. **Evaluation:**

After training the neural network on the training data, you need to evaluate its performance using the test data. The evaluation process measures how well the model generalizes to unseen data and provides insights into its accuracy and effectiveness.

3.1.6. **Image Prediction:**

Once the neural network model is trained, it can take new hand gesture images as input and make predictions about which letter from 'A' to 'I' the hand gesture represents.

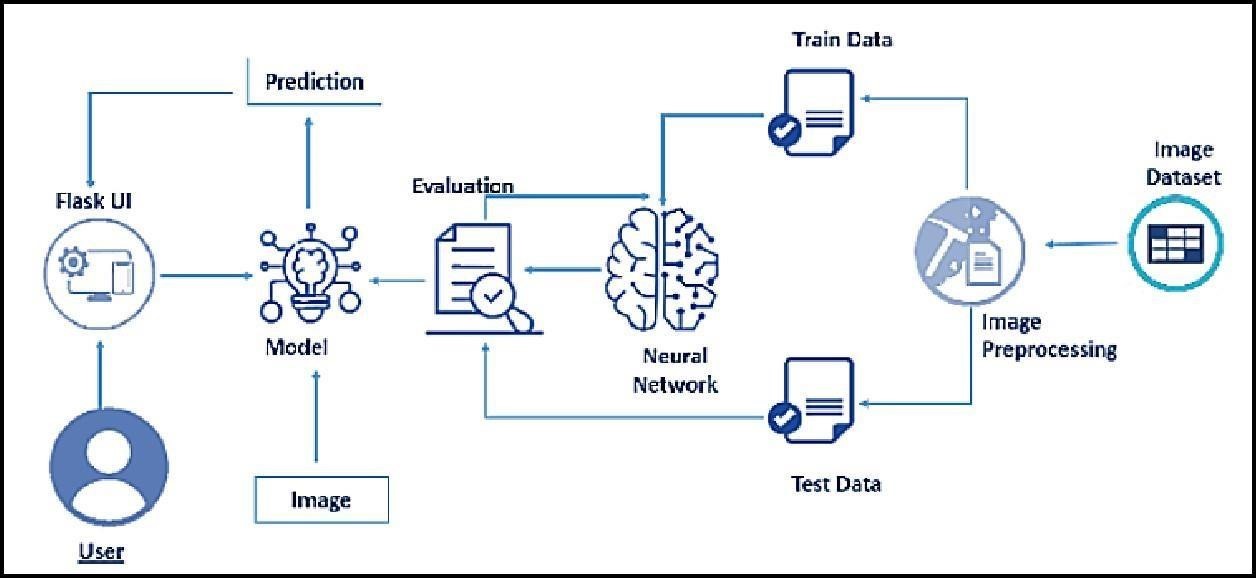
3.1.7. **Flask UI (User Interface):**

Flask is a web framework for Python, and in our project, it serves as the user interface (UI). The Flask UI allows users to interact with the trained model through a web application. Users can

upload or capture hand gesture images through the UI and receive predictions from the model in response.

3.1.8. **User:**

The end-users of our project are individuals who want to communicate using hand gestures. They interact with the Flask UI to input their hand gesture images and obtain predictions for the corresponding letters.



**Fig. 3. 1. Block Diagram**

**3.2 HARDWARE / SOFTWARE REQUIREMENTS**

3.2.1 **Hardware Requirements:**

**1.Processor**: Intel CoreTM i5-9300H with a clock speed of 2.4GHz.

**2.RAM Size**: 8GB DDR (Random Access Memory) to handle data and computations efficiently.

**3.System Type**: X64-based processor architecture for compatibility and performance.

**4.Webcam**: An integrated or external webcam with Full HD support for capturing high-quality images and video feed.

3.2.2 **Software Requirements:**

**1.Desktop GUI**: Anaconda Navigator, which provides an integrated development environment (IDE) for data science tasks and managing Python environments.

**2.Operating System**: Windows 10, as the base operating system for running the project.

**3.Front-End**: HTML and CSS will be used for developing the user interface and enhancing user experience.

**Programming Language**:

1.Python, which serves as the primary language for implementing the fuel efficiency prediction system using machine learning algorithms.

By utilizing the above hardware and software components, the project can be efficiently developed, offering a user-friendly interface and accurate fuel efficiency predictions for various vehicles.

**4 EXPERIMENTAL INVESTIGATIONS**

**4.1 Visualizing and Analysing the Data**

4.1.1 **Importing the Libraries**: In this code snippet, we import the necessary libraries and modules required for building and training a deep learning model for image classification. The TensorFlow library provides the foundation for creating and training the neural network, while OpenCV is used for image processing tasks. Other modules from TensorFlow's Keras API are imported to construct the model architecture and handle image data.

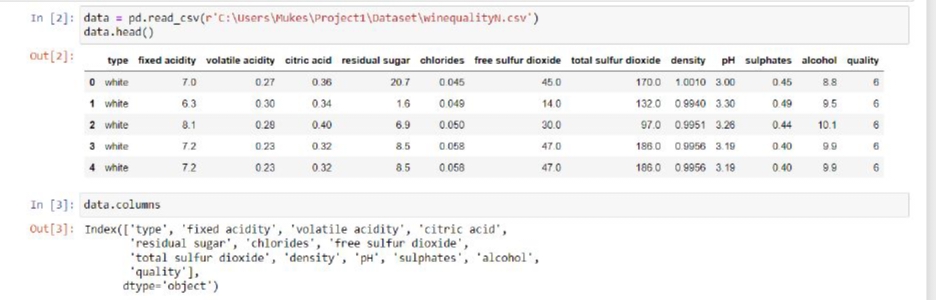


4.1.2 **Read the Dataset**

Our dataset format might be in .csv, excel files, .txt, .json, etc. We can read the dataset with the help of pandas.

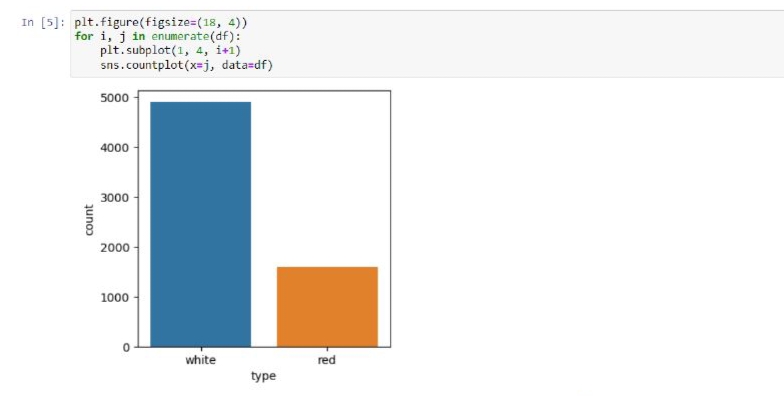
In pandas we have a function called **read\_csv()** to read the dataset. As a parameter we must give the directory of csv file.

These are the different columns present in our dataset, let’s understand what these features indicates:



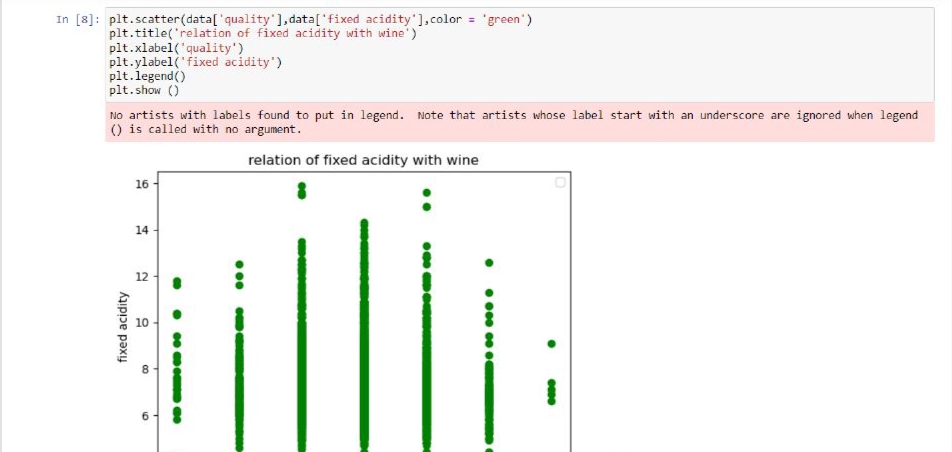
### 4.1.3**Univariate Analysis**

### In simple words, univariate analysis is understanding the data with a single feature. Here we have displayed two different graphs such as pie plot, box plot and count plot.



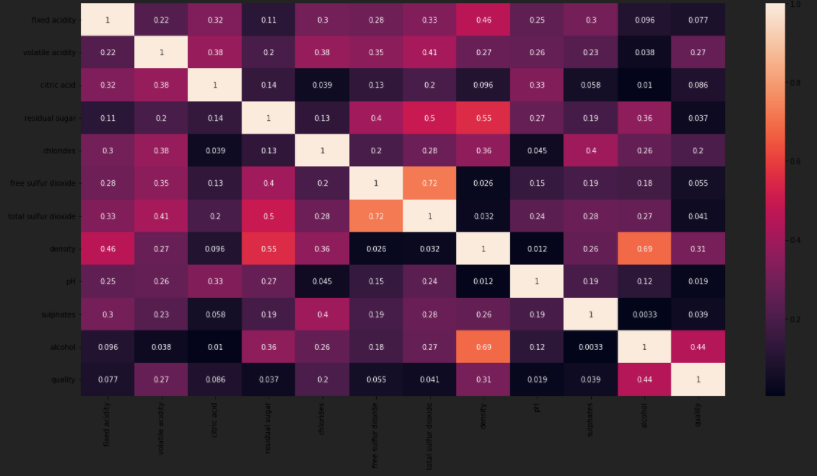
### 4.1.4 **Bivariate Analysis**

To find the relation between two features we use bivariate analysis.



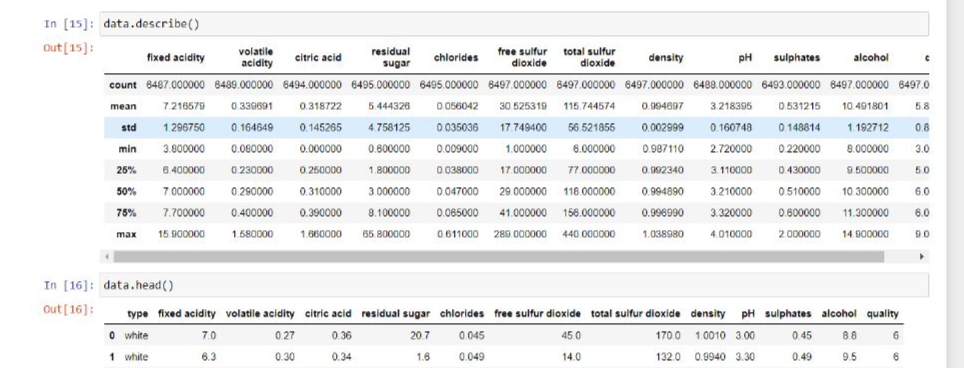
4.1.5 **Multivariate Analysis**

In simple words, multivariate analysis is to find the relation between multiple features. Here we have used a heatmap from the seaborn package.



### 4.1.6 **Descriptive Analysis**

Descriptive analysis is to study the basic features of data with the statistical process. Here pandas have a worthy function called describe. With this describe function we can understand the unique, top and frequent values of categorical features. And we can find mean, std, min, max and percentile values of continuous features.



### **4.2 Data Pre-Processing**

As we have understood how the data is. Let's pre-process the collected data.The download data set is not suitable for training the machine learning model as it might have so much randomness, so we need to clean the dataset properly to fetch good results. This activity includes the following steps.

4.2.1 **Drop Unwanted Features**

drop () is used to drop specified labels from rows or columns.

Remove rows or columns by specifying label names and corresponding axis, or by specifying directly index or column name

### 4.2.2 **Checking for Null Values**

For checking the null values, df.isnull() function is used.

### 4.2.3 **Handling Categorical Values**

As we can see our dataset has categorical data, we must convert the categorical data to integer encoding or binary encoding.

To convert the categorical features into numerical features we use encoding techniques. There are several techniques but, in our project, we are using feature mapping and label encoding.

### 4.2.4 **Splitting Data into Train and Test**

Now let’s split the Dataset into train and test sets. First split the dataset into x and y and then split the data set

### 4.2.5 **Feature Scaling**

Feature scaling is a method used to normalize the range of independent variables or features of data. Standard scaler () is initialized. Independent training data is passed in the fit\_transform() method and independent test data is passed in the transform () function.

**4.3 Model Building**

Now our data is cleaned and it’s time to build the model. We can train our data on different algorithms. For this project we are applying two regression algorithms. The best model is saved based on its performance.

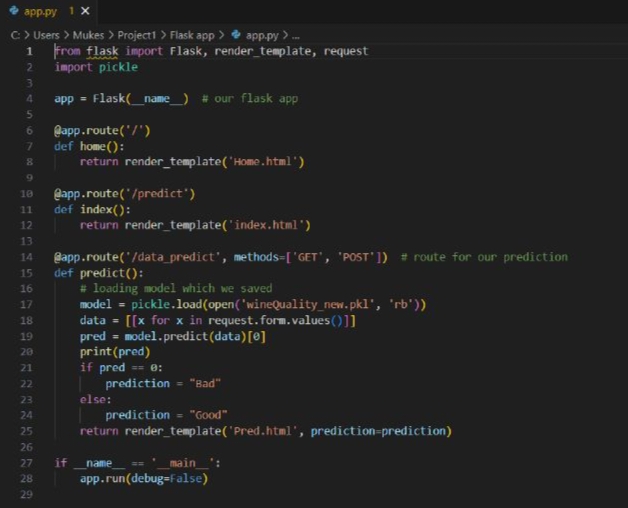


**4.4 Application Building**

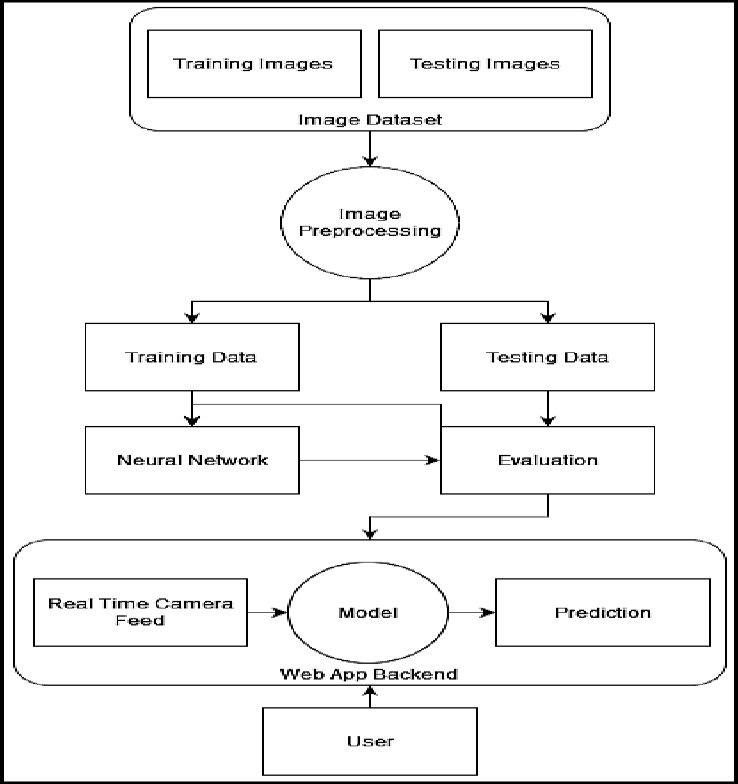
This section will build a web application integrated into the model we built. A UI is provided for the users where he must enter the values for predictions. The enter values are given to the saved model and prediction is showcased on the UI.

This section has the following tasks

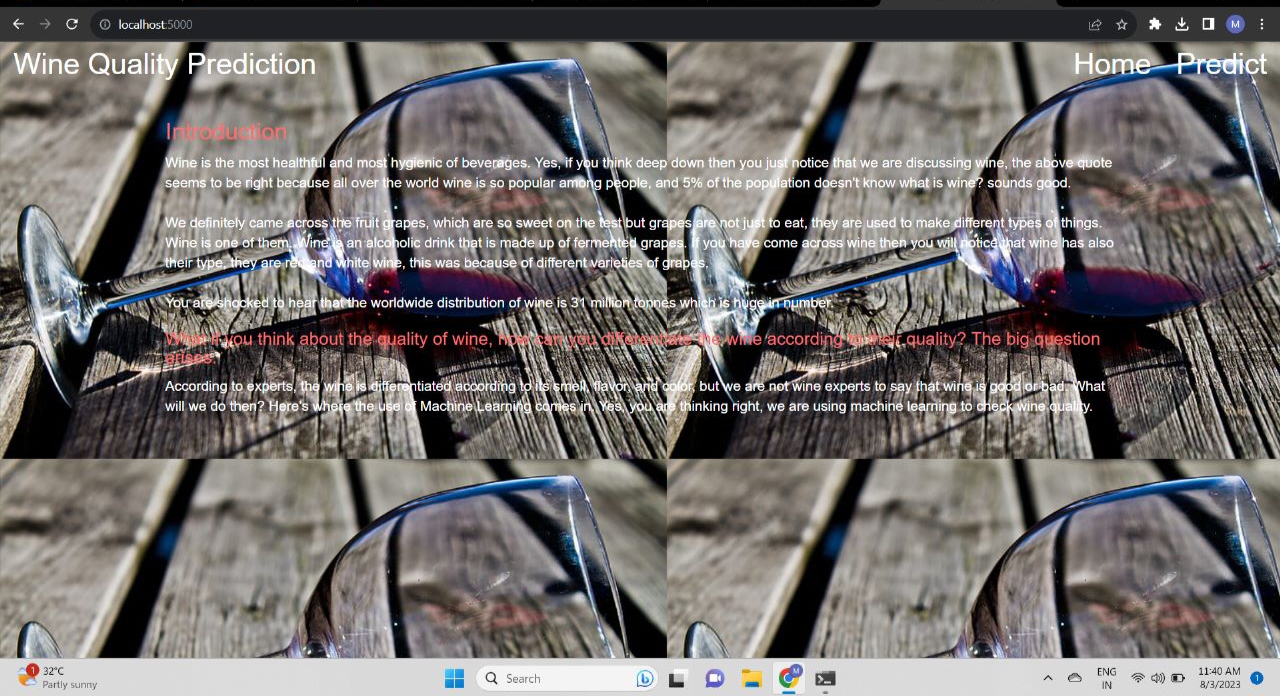
* Building HTML Pages
* Building server-side script



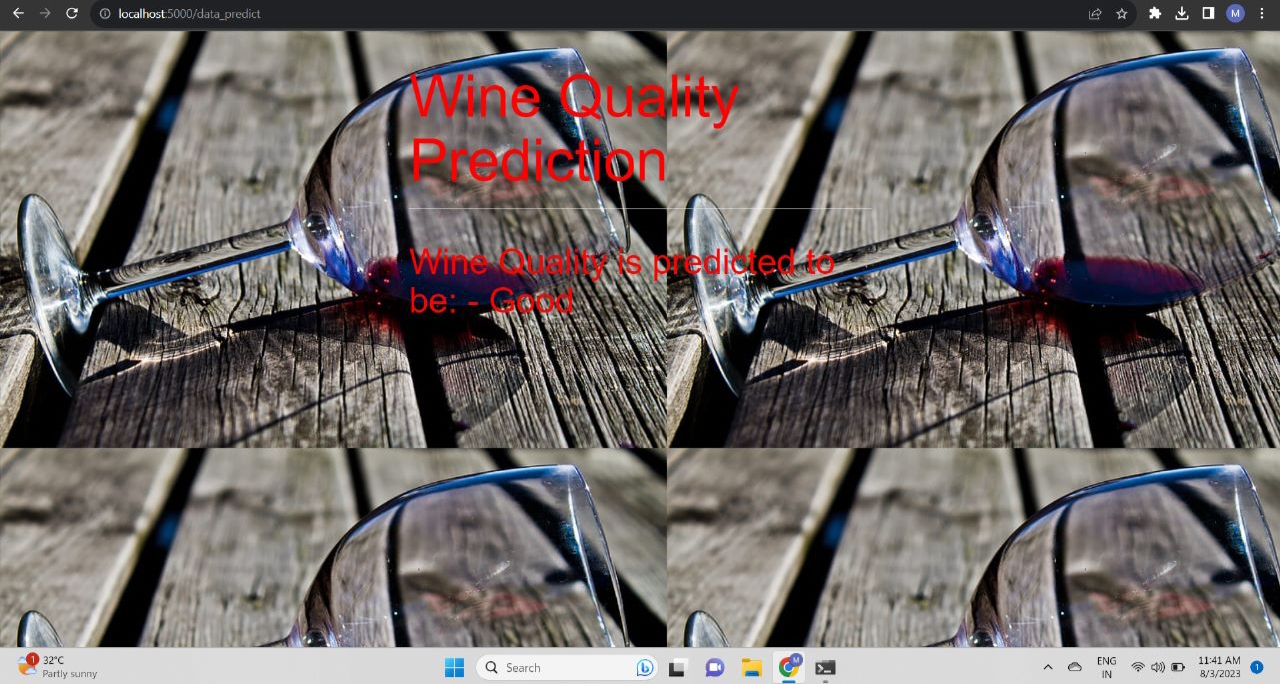
**5.** **FLOWCHART**



**6. Result**







**7. ADVANTAGES & DISADVANTAGES**

**ADVANTAGES:**

Objective Assessment: The proposed solution offers an objective and data-driven approach to assess wine quality, reducing subjective biases and human errors.

Efficiency: By automating the prediction process, winemakers can save time and resources in evaluating wine quality, allowing them to focus on other aspects of production.

Personalized Recommendations: Wine enthusiasts can discover wines that match their preferences based on predicted quality scores, enhancing their wine selection experience.

Scalability: Once the model is trained, it can efficiently predict the quality of a large number of wine samples, making it scalable for use in the wine industry.

**DISADVANTAGES:**

Reliance on Data Quality: The effectiveness of the model heavily depends on the quality, representativeness, and completeness of the training dataset. Poor quality data can lead to inaccurate predictions.

Overfitting: If the model is overly complex or not properly regularized, it may suffer from overfitting, resulting in poor generalization to new data.

Feature Selection: Selecting relevant features from the vast array of chemical and sensory attributes requires domain knowledge and may introduce biases if not done carefully.

Interpretability Challenges: Some complex machine learning algorithms may lack interpretability, making it challenging to explain model predictions to stakeholders.

Domain Specificity: The model's effectiveness might be limited to the specific wine dataset used for training, and it may not generalize well to other wine varieties or regions.

**8. APPLICATIONS**

The solution for predicting the quality of wine using machine learning has various applications across different domains. Some of the key areas where this solution can be applied include:

Winemaking Industry:

In the winemaking industry, the predictive model can be integrated into the production process for quality control and improvement. Winemakers can use the model to assess the quality of wine samples during fermentation, aging, and blending, enabling them to make informed decisions to enhance the overall quality of the final product.

Wine Retail and E-commerce Platforms:

Online wine retailers and e-commerce platforms can employ the predictive model to provide personalized wine recommendations to customers. By understanding the preferences of individual customers, the platform can suggest wines that align with their taste preferences and quality expectations, leading to improved customer satisfaction and retention.

Sommeliers and Wine Professionals:

**9. CONCLUSION**

In conclusion, the project aimed to predict the quality of wine using machine learning techniques based on its chemical composition and sensory attributes. Through a comprehensive analysis and implementation, the following key findings and contributions were made.

**10. FUTURE SCOPE**

Incorporating Advanced Deep Learning Architectures: Currently, the project focuses on traditional machine learning algorithms. In the future, exploring advanced deep learning architectures, such as Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs), may improve feature extraction and prediction performance.

Transfer Learning: Leveraging transfer learning can be beneficial when dealing with limited labeled data. Pre-trained models from related domains, such as food or beverage quality prediction, can be fine-tuned for wine quality prediction, saving computation time and improving generalization.

Ensemble Methods: Implementing ensemble methods like bagging or boosting, where multiple models are combined, can further enhance prediction accuracy and robustness.

**11. BIBILOGRAPHY**

Cortez, P., & Silva, A. (2009). Modeling wine preferences by data mining from physicochemical properties. Decision Support Systems, 47(4), 547-553.

Quinlan, J. R. (1993). C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. Annals of Statistics, 29(5), 1189-1232.

**12. APPENDIX**

**Source code**

This the source code link given below:

https://github.com/MukeshKethe/Externship\_Project\_GEC