

**G H Patel College of Engineering & Technology**

**(A Constituent College of CVM University)**

**New V. V. Nagar**

**COMPUTER ENGINEERING DEPARTMENT**

**AIML Report**

**on**

***Wine Quality Prediction***

**Submitted By**

**Name of Student :** Prushti Lathiya

**Enrolment Number :** 12302130503005

**Name of Student :** Kalp Patel

**Enrolment Number :** 12302130503006

**Guided By**

**Prof. Priyang Bhatt**

**Artificial Intelligence & Machine Learning (202046702)**

**A.Y. 2024-25 EVEN TERM**

**Title of the Project:**

* **Wine Quality Prediction**

**Objective:**

The primary objective of this project is to develop a machine learning model capable of accurately predicting the quality of wine based on its physicochemical properties. By leveraging a Random Forest classification algorithm, the project aims to:

* Analyze and understand the relationship between various chemical features of red and white wines (such as acidity, sugar content, alcohol level, pH, etc.) and their assigned quality scores.
* Build a robust predictive model using the Random Forest algorithm that can generalize well on unseen wine data.
* Provide valuable insights into which chemical features most influence wine quality, supporting potential decision-making in wine production and quality control.
* Create a modular, scalable, and reusable ML pipeline that follows standard industry practices with structured code, documentation, and version control.
* Generate a comprehensive evaluation report using metrics like accuracy, precision, recall, F1-score, and confusion matrix to assess model performance.
* The final deliverable is a complete machine learning solution that includes preprocessing, training, model evaluation, and reporting — organized using a professional project structure and uploaded to a GitHub repository for transparency and collaboration.

**Dataset Used:**

1. **winequality-red.csv** – Contains physicochemical data for red wines
2. **winequality-white.csv** – Contains the same type of data for white wines

These datasets were merged and processed to build a unified model that can predict the quality of a wine sample based on its chemical properties.

**Key Features in the Dataset:**

|  |  |
| --- | --- |
| **Feature Name** | **Description** |
| fixed acidity | Tartaric acid concentration (g/dm³) |
| volatile acidity | Acetic acid concentration – too much can lead to an unpleasant taste |
| citric acid | Citric acid concentration – adds freshness and flavor |
| residual sugar | Amount of sugar remaining after fermentation (g/dm³) |
| chlorides | Salt concentration (g/dm³) |
| free sulfur dioxide | SO₂ freely available to prevent microbial growth (mg/dm³) |
| total sulfur dioxide | Total SO₂ concentration (free + bound) |
| density | Density of the wine (g/cm³) |
| pH | pH level (acidity/basicity) |
| sulphates | Sulfate concentration – affects wine preservation and flavor |
| alcohol | Alcohol percentage by volume (%) |
| quality (target) | Score between 0 and 10 given by wine tasters (used as target variable) |

**Model Chosen:**

**Why Random Forest?**

Random Forest is an ensemble learning method that builds multiple decision trees and combines their outputs for better accuracy and robustness. It was chosen because:

* It handles **non-linear relationships** and **high-dimensional data** well
* It provides **feature importance**, helping interpret the influence of each variable
* It is **resistant to overfitting** due to its ensemble nature
* It works well with **classification tasks** like predicting discrete wine quality scores

**Model Details:**

* **Type:** Supervised Classification
* **Algorithm:** Random Forest (Ensemble of Decision Trees)
* **Library Used:** sklearn.ensemble.RandomForestClassifier
* **Hyperparameters Used:**
  + n\_estimators = 100 – Number of trees in the forest
  + max\_depth = None – Nodes are expanded until all leaves are pure
  + random\_state = 42 – Ensures reproducibility

The model was trained on preprocessed features and evaluated using metrics like **accuracy**, **precision**, **recall**, **F1-score**, and a **confusion matrix**.

**Performance Matrics:**

**Evaluation Metrics Used:**

|  |  |
| --- | --- |
| **Metric** | **Description** |
| **Accuracy** | Proportion of total predictions that were correct |
| **Precision** | Ability of the classifier to not label a negative sample as positive |
| **Recall** | Ability of the classifier to find all the positive samples |
| **F1-Score** | Harmonic mean of precision and recall |
| **Confusion Matrix** | A table showing true vs. predicted classifications for all classes |

**Example:**

Assuming the model predicts wine quality scores from 3 to 8, here's a sample confusion matrix (simplified):

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Actual \ Predicted** | **3** | **4** | **5** | **6** | **7** | **8** |
| **3** | 10 | 2 | 0 | 0 | 0 | 0 |
| **4** | 3 | 25 | 5 | 0 | 0 | 0 |
| **5** | 0 | 6 | 130 | 15 | 1 | 0 |
| **6** | 0 | 0 | 12 | 145 | 20 | 0 |
| **7** | 0 | 0 | 1 | 14 | 80 | 5 |
| **8** | 0 | 0 | 0 | 1 | 3 | 12 |

**Challenges & Learnings:**

**Challenges:**

1. **Data Cleaning & Integration**
   * Merging two datasets (red and white wine) required consistent formatting and handling of different distributions in feature values.
   * The CSV files used (;) as separators, which initially caused issues while reading the data correctly.
2. **Imbalanced Classes**
   * The quality scores were not evenly distributed — most wines had quality ratings between 5 and 7, making the classification imbalanced and challenging for the model to distinguish between minority classes.
3. **Model Selection**
   * Choosing an appropriate model involved trying multiple classifiers before finalizing Random Forest based on performance and interpretability.
4. **Overfitting Concerns**
   * During training, ensuring the model didn’t memorize the data but could generalize well on unseen data required careful validation.
5. **Project Structure & Modularity**
   * Organizing the project in a modular format (with folders like src/, models/, docs/, etc.) was initially challenging but essential for clarity and scalability.

**Learnings:**

1. **Importance of Preprocessing**
   * Clean, consistent, and well-scaled data significantly improves model performance.
2. **Model Evaluation Metrics Matter**
   * Using multiple evaluation metrics (accuracy, precision, recall, F1-score) gives a clearer picture than just relying on accuracy alone.
3. **Feature Importance Insights**
   * Random Forest helped identify which chemical features (like alcohol, sulphates, volatile acidity) have the strongest impact on wine quality.
4. **Project Workflow Discipline**
   * Following a professional project structure, writing reusable code, and generating automatic reports improved the maintainability and reproducibility of the project.
5. **Version Control with GitHub**
   * Using GitHub for project backup, version control, and collaboration is a powerful industry-standard practice.

**Tools & Libraries:**

**Programming Language:**

* **Python 3.x** – The primary language used for building the machine learning pipeline due to its simplicity and rich ecosystem of ML libraries.

**Key Libraries:**

|  |  |
| --- | --- |
| **Library** | **Purpose** |
| **pandas** | For loading, handling, and preprocessing the CSV datasets |
| **numpy** | For efficient numerical operations and array manipulation |
| **matplotlib** | For creating plots and graphs |
| **seaborn** | For statistical data visualization and heatmaps |
| **scikit-learn** | For model building, training, preprocessing, and evaluation |
| **joblib** | For saving and loading the trained Random Forest model |

**Tools Used:**

|  |  |
| --- | --- |
| **Tool** | **Purpose** |
| **Jupyter Notebook** | Interactive environment for writing and running ML code |
| **Git** | Version control for project source code |
| **GitHub** | Hosting the project repository and collaboration |