Predicting Wine Quality using Machine Learning

Course: Artificial Intelligence/Machine Learning

Assignment SuSe 2024

By:

Alkistis Sytilidou #3120096

Amina Kameric #3114464

Erion Vathi #3114520

Fawziah Alali #3117897

Faris Alali #3117901

Table of Contents

Introduction

This project aims to predict the quality of wine based on its chemical properties. We used the UCI Wine Quality dataset from Kaggle, which includes various chemical measurements of wine samples and their quality ratings. The primary objective is to build a model that can accurately predict wine quality and evaluate its performance using various metrics.

i. Project Repository:

The source code for this project, including all scripts, data preprocessing steps, model training, and evaluation, can be found on GitHub at the following link:
https://github.com/alkistissyt/Wine_quality_prediction/tree/main

ii. Dataset description:

The dataset contains 1,143 samples of red wine with 12 features each. The target variable is the quality rating, which ranges from 3 to 8.

The features include:

Fixed acidity, Volatile acidity, Citric acid, Residual sugar, Chlorides, Free sulfur dioxide,

Total sulfur dioxide, Density, pH, Sulphates , Alcohol

Dataset url: https://www.kaggle.com/datasets/yasserh/wine-quality-dataset/data

Data Preprocessing

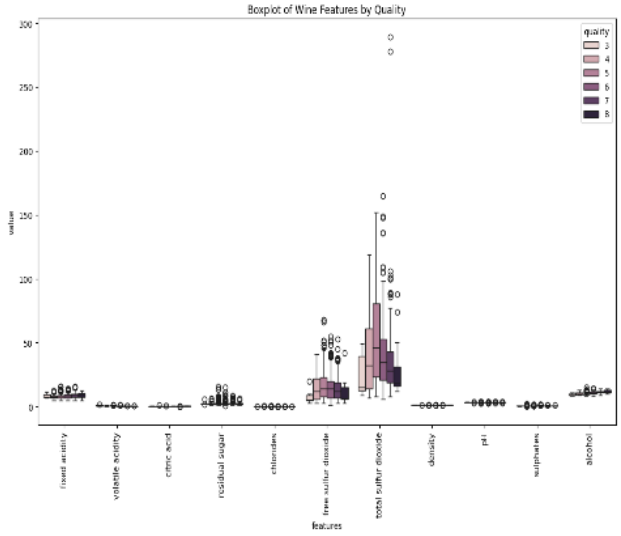
i. Feature Selection:

The id column was identified and dropped, as it did not provide any useful information for the analysis, leaving the dataset shape as (1143, 12).

ii. Checking/Handling Missing Values

An initial inspection was performed to check for any missing or null values using the describe() and info() functions. The output of these summary statistics indicated that there are no missing or null values in any of the columns, as the count of 1143.000000 remains the same across all columns.

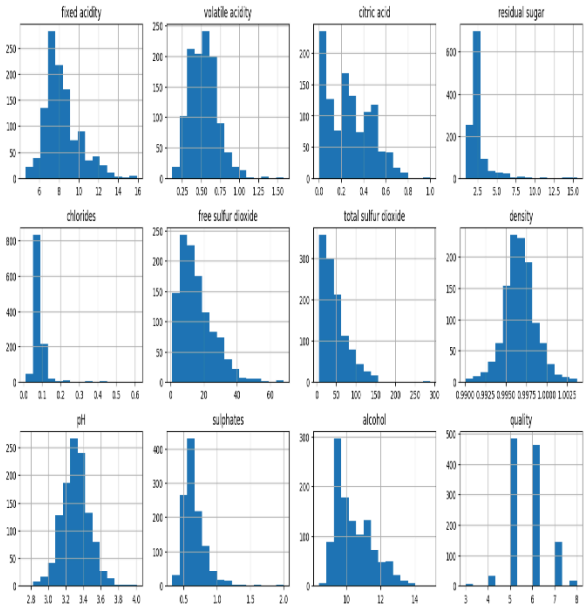
Exploratory Data Analysis (EDA)



i. Boxplot

1. **Sulfur Dioxide Levels:** In the box plot, the most visually distinctive pattern is the levels of sulfur dioxide. This is because the measurement unit for the compound's amount makes the value have a high number when compared to other units like pH, density, or alcohol.
2. **Outliers**: There are many outliers in each variable we are analyzing. Later on it would be beneficial to remove them from the training data in order to improve the AI models accuracy.

ii. Histograms



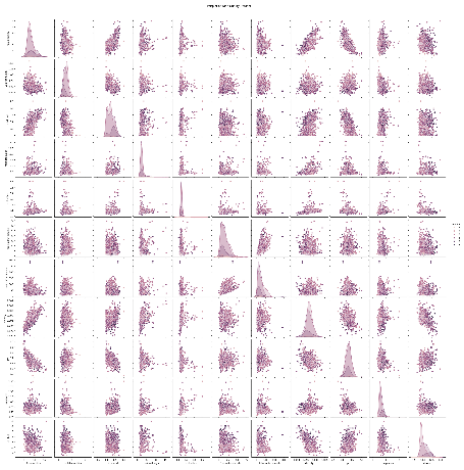
Extraordinary Observations

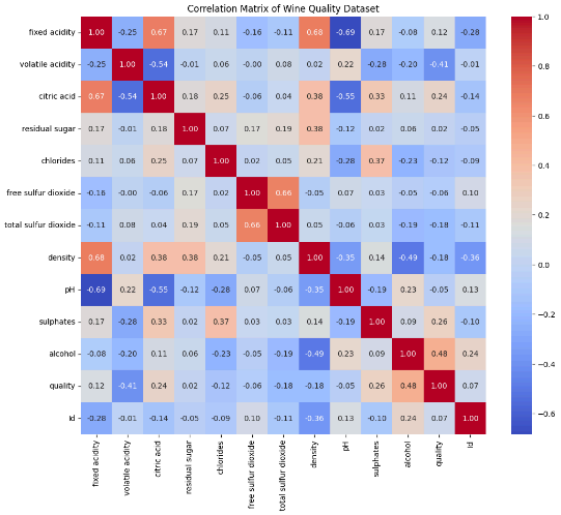
1. **Residual Sugar and Chlorides**: Both features show highly right-skewed distributions, indicating that most wines have low levels of residual sugar and chlorides, but there are a few wines with much higher values.
2. **Free and Total Sulfur Dioxide**: Both features show a right-skewed distribution, with most values being relatively low. However, there are some wines with significantly higher sulfur dioxide levels.
3. **Quality**: The quality ratings are discrete and show a clear peak at 5, 6, and 7. There are very few wines with ratings below 5 or above 7.

The y-axis represents the number of samples with those acidity values.

The y-axis represents the number of samples with those values.

iii. **Pairplot and Heatmap**





1. **Alcohol and Quality**: positive correlation
2. **Sulfates and Quality**: positive correlation
3. **Volatile Acidity and Quality**: negative correlation

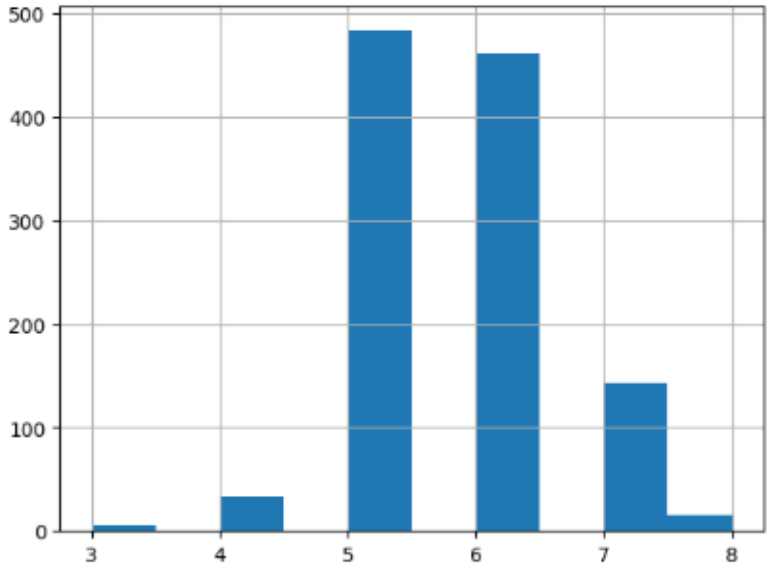
Model Training

Classification is a type of supervised learning where the goal is to assign labels to instances based on their features. In our case, the wine quality ratings are categorical labels that we want to predict. The quality of wine was originally rated on a scale from 3 to 8, but for better interpretability and to simplify the problem, we categorized these ratings into three labels: "low", "mid", and "high". This transforms our problem into a classification task where the goal is to assign one of these three labels to each wine sample.

The Random Forest classifier, which we used for this task, is a classification algorithm. It builds multiple decision trees, in our case 100 decision trees (`n_estimators= 100`) during training and outputs the most frequent class (mode) of the individual trees for classification.

After training the model on the training data, we use it to predict the quality labels for the test data. The model uses the learned patterns from the training data to make predictions on the unseen test data.

i. Quality Labeling



Given the distribution of the `quality` ratings, we decided to categorize the `quality` into three labels: "low", "mid", and "high". This categorization helps simplify the classification task.

We used the following mapping to create the `quality_label` column: - `3` and `4`: "low" - `5` and `6`: "mid" - `7` and `8`: "high"

ii. Data splitting

To evaluate the performance of the machine learning models, the dataset was split into training and testing sets. This approach allows the model to be trained on one subset of the data and tested on another, ensuring that the model's performance generalizes well to unseen data.

The target variable (`y`) is the `quality_label` and with the `train_test_split` function from `scikit_learn` was used to split the data into 80% training and 20% testing sets to ensure the model has sufficient data to learn from while keeping a reasonable portion for evaluation.

After splitting the data, we obtained the following shapes for the training and testing sets:

Training Set:

```
X_train shape: (914, 12)
y_train shape: (914,)
```

Testing Set:

```
X_test shape: (229, 12)
y_test shape: (229,)
```

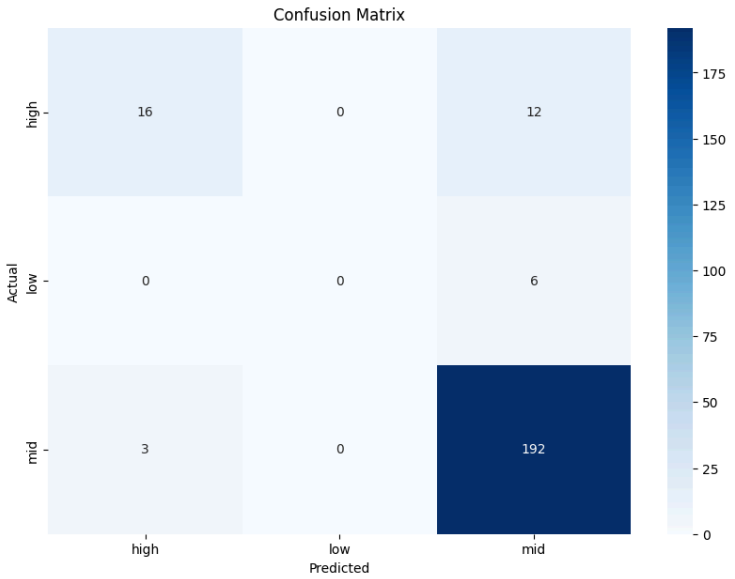
Model Evaluation

i. Accuracy Score:

Accuracy is a common metric for classification problems. It represents the ratio of correctly predicted instances to the total instances in the dataset.

ii. Confusion Matrix Analysis:

The confusion matrix provides a more detailed breakdown of the model's performance. It shows the number of correct and incorrect predictions for each class, helping us identify where the model is making errors.



Key Observations

1. **High Accuracy for 'mid' Quality**:

   The model has a very high true positive rate for predicting 'mid' quality wine (192 correct predictions).

2. **Struggles with 'low' Quality**:

   The model did not predict any instance as 'low' correctly, indicating a potential issue in distinguishing 'low' quality wines.

   This is also seen though the unique values:

```
Unique values in y_test before mapping: ['mid' 'high' 'low']
Unique values in y_pred before mapping: ['mid' 'high']
```

Addressing Class Imbalance

i. Class Distribution:

The class distribution in the training, testing, and predicted datasets is as follows:

```
Class distribution in y_train:
quality_label
mid     750
high    131
low      33

Class distribution in y_test:
quality_label
mid     195
high     28
low       6

Class distribution in y_pred:
mid     210
high     19
```

From these statistics, it is evident that the 'mid' class dominates the dataset, while the 'low' class is significantly underrepresented. This imbalance is likely affecting the model's ability to accurately predict 'low' and 'high' quality wines, as seen in the predictions where the 'low' class is entirely missing.

During model training, class weights can be adjusted to give more importance to the minority classes, in the Random Forest classifier, this can be achieved by setting the parameter to `class_weight='balanced'`

Some further potential solution entail:

***Oversampling the Minority Class**:*

→Use techniques such as SMOTE (Synthetic Minority Over-sampling Technique) to generate synthetic samples for the minority class.

→This helps in balancing the dataset by increasing the number of instances in the minority class.

Or even Undersampling the Majority Class:

→ Randomly remove samples from the majority class to balance the dataset.

→ This approach can be useful when the majority class significantly outnumbers the minority class.

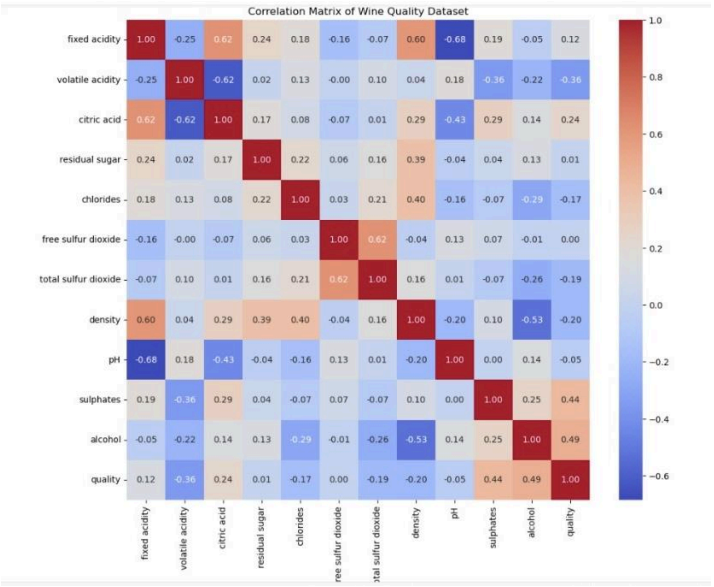
(***A Hybrid method could entail a mix of both above for a balance)***

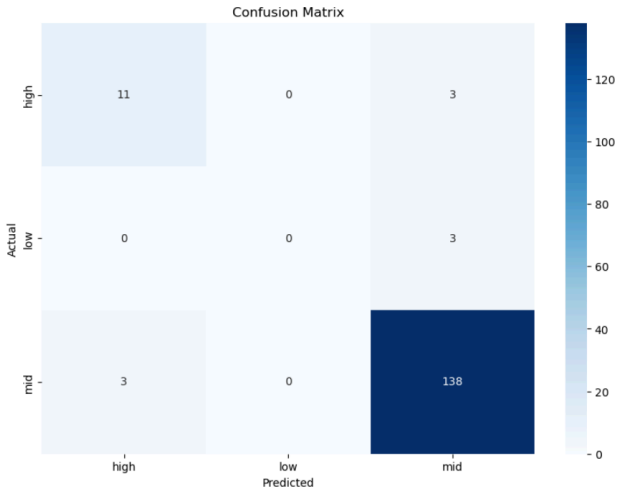
Model Tuning and Optimization

Removing all outliers

A quick, but data limiting method that can be used for tuning the model is removing all outliers in the dataset. Outliers are extreme values that differ significantly from the majority of the data. Identifying and removing these outliers can help improve the accuracy of the model, as they often are anomalies that can skew the results. The problem with this method however is that each column has its own set of outliers, meaning that if one outlier is found in any of the columns, the entire row containing it needs to be removed.

This will significantly reduce the amount of data we have, at the benefit of slightly improving accuracy.





The dataset shape went from 1143 rows to 629 rows, almost half of what the original dataset was, however the accuracy is 0.94, which is 0.03 more than the bigger dataset (0.91). It seems like removing all outliers from every column results in overfitting. In further development, managing outliers more accordingly can give worthwhile trades of data amount for better accuracy.

Other tuning/optimization methods include:

Addressing Class Imbalance: In class-imbalanced datasets, when a particular class is substantially underrepresented, Implementing techniques such as oversampling the minority class or using synthetic data generation methods (e.g., SMOTE) will ensure a more balanced training dataset.

Model Tuning and Optimization: Experiment with different hyperparameters and classifiers to improve prediction accuracy, particularly for the 'low' quality class.

Feature Engineering: Explore additional feature engineering techniques to extract more relevant information from the existing features

.

Conclusion

In this project, we successfully developed a machine learning model to predict the quality of wine based on its chemical properties. Using the Random Forest classifier, we achieved a high accuracy rate of 0.91. Our exploratory data analysis highlighted significant trends and correlations among the features, which informed our model training. However, the model struggled to predict 'low' quality wines accurately due to class imbalance.

Future Work

Future improvements can include:

→ Suggestions for the Class Balance (implementing such described techniques)

→ Model Tuning and Optimization: Experimenting with different hyperparameters and classifiers to improve prediction accuracy, particularly for the 'low' quality class.

→Outlier Managing: Maninging outlier amount and maintaining it to a certain level where data amount is not hindered.

References

https://www.kaggle.com/code/rahafrashed/wine-quality-prediction

https://www.kaggle.com/code/mohamedkhaledmahmoud/red-wine-quality-prediction-model

https://www.kaggle.com/code/aditishelke19/classification-wine

https://www.kaggle.com/code/sriramyaattuluri/3-red-wine-quality-analysis

https://www.geeksforgeeks.org/getting-started-with-classification/