

Red Wine Quality

Marten Tiisler

Tanel Tooming

Sven-Erik Taru

GitHub repository: <https://github.com/taneltooming/RedWineQuality>

Business understanding

- **Identifying your business goals**

- **Background**

Wine quality plays a key role in both consumer satisfaction and producer decision-making. Traditional quality assessment requires human experts, which is slow, subjective, and costly. We aim to create a more objective approach using chemical measurements to estimate wine quality. Not only does this make the whole process quicker, but also cheaper.

- **Business goals**

The main goal is to increase the consistency and objectivity of wine quality evaluation with a machine learning model that can predict the quality of wine based on given chemical properties. This can help save money and time.

The secondary goal is to identify which features have the strongest influence on the quality, as this can help winemakers make the necessary changes.

- **Business success criteria**

The predictive model should achieve an AUC score of at least 0.85. The model should also achieve a balanced precision-recall ratio to ensure that valuable products are not overlooked. The outcome should be presented in a way that is understandable to everybody.

- **Assessing your situation**

- **Inventory of resources**

Public Kaggle dataset “Red Wine Quality”, consisting of 1599 samples with 11 chemical features and the quality label.

Python environment Jupyter Notebook with the libraries needed to train our model and showcase the results.

Github repository to save our code and document our project.

Three team members, each with their laptops.

- **Requirements, assumptions, and constraints**

The requirements are that the code is reproducible and our work is well documented. If code is taken from the internet, we must credit the original creators. All work must be completed within the project deadlines.

The assumption is that the dataset accurately represents real wine quality measurements with realistic correlations. We also assume that the quality labels in the dataset are correct.

The main constraint is that the dataset is relatively small.

- **Risks and contingencies**

Overfitting is a risk due to limited dataset size. However, we also have access to a much bigger white wine quality dataset with the exact same features, so the solution could be to use them both and compare their results.

- **Terminology**

Quality score - integer rating from 0–10 that describes the quality of the wine

Fixed acidity - amount of acids that do not evaporate easily, therefore they play a big part in the wine's fundamental sourness

Volatile acidity - the amount of acetic acid in wine, which at too high of levels can lead to an unpleasant, vinegar taste

Citric acid - found in small quantities, citric acid can add 'freshness' and flavor to wines

Residual sugar - the amount of sugar remaining after fermentation stops

Chlorides - the amount of salt in the wine

Free sulfur dioxide - free form of dioxide gas (SO₂) that exists in equilibrium between molecular SO₂ and bisulfite ion that prevents oxidation and microbial spoilage

Total sulfur dioxide - amount of free and bound forms of SO₂

Density - similar to that of water. Concentration of alcohol lowers density and dissolved sugars increase it

pH - describes how acidic or basic a wine is on a scale from 0 (very acidic) to 14 (very basic)

Sulphates - a wine additive which can contribute to sulfur dioxide gas (SO₂) levels

Alcohol - the percent alcohol content of the wine

SVM RBF - A Support Vector Machine that uses a Radial Basis Function (RBF) kernel that measures similarity between data points based on their distance to separate classes by modeling nonlinear relationships in the data

Random Forest - A model that combines many randomly built decision trees and outputs their majority vote or average prediction

XGBoost - An algorithm that builds decision trees sequentially, with each new tree correcting the errors of the previous ones

AUC score - A performance metric that measures how well a model separates classes

- **Costs and benefits**

There are no costs in terms of money, as the dataset is public and training the models is free. The only cost is the time spent on completing the project. The biggest benefit is the cost-saving potential, because from a clients' point of view the model would provide a cheap and quick alternative to paying for long and expensive tests. The project also provides an improved understanding of key wine-quality factors, which is very useful knowledge for wine-makers who want to improve the quality of their product.

- **Defining your data-mining goals**

- **Data-mining goals**

We will train 3 different models (SVM RBF, Random Forest and XGBoost) using the preprocessing steps that suit them the most. The goal is to train them to the best of their ability and compare their results. We will try to understand which chemical features contribute most to the classification

outcome. Eventually we will produce visualizations and reports illustrating relationships, feature importance, and model performance.

- **Data-mining success criteria**

Achieve an AUC score of 0.85 on at least 1 model.

Providing reproducible, well documented code and interpretable explanations of feature importance from the viewpoint of a person who is not familiar with our project.

Data understanding

- **Gathering data**

- **Outline data requirements**

A dataset with sufficient sample size (1000+ rows).

The dataset contains wine samples with objective physiochemical measurements along with an evaluation of wine quality. Each row represents a unique sample:

- independent numeric features (chemical measurements), which may influence wine quality
- the target variable - quality score (0-10)

The required time range does not matter, as only the chemical properties play a role in the quality rating and they do not necessarily change depending on the year.

- **Verify data availability**

The Red Wine Quality dataset is a publicly accessible dataset from the UCI Machine Learning Repository and it has been shared to Kaggle for convenience. The dataset includes:

- 1599 samples
- 11 input variables
- 1 output variable

Due to privacy restrictions, the data doesn't include brand, price, grape type, origin sub-region nor production method. But that doesn't change the fact that the data is available for both regression and classification tasks. All of the data we want is available.

- **Define selection criteria**

We decided to use:

- only the red wine dataset
- all samples and variables provided in the dataset

- **Describing data**

The Red Wine Quality dataset contains (first 11 being numerical input variables based on physiochemical tests and the 12th one being the output variable based on sensory data):

1. fixed acidity
2. volatile acidity
3. citric acid
4. residual sugar

5. chlorides
6. free sulfur dioxide
7. total sulfur dioxide
8. density
9. pH
10. sulphates
11. alcohol
12. quality (score between 0 and 10)

- **Exploring data**

When we explored the dataset, we first checked how each feature is distributed. Almost all features stay within normal chemical ranges and there are no missing values. A few features (mainly volatile acidity, residual sugar, chlorides and sulphates) have some higher values, but these look like natural outliers rather than mistakes. However, the feature “total sulfur dioxide” includes two very high outliers, which are above typical limits for red wine. These values are possible outliers.

The target variable quality is clearly imbalanced. Most wines are rated 5 or 6, while very high or very low scores are rare. This means that some models may end up predicting the majority classes more often, so we will need to keep this in mind during modelling.

Looking at simple correlations, alcohol shows the strongest positive relationship with quality, while volatile acidity has one of the strongest negative relationships. These patterns seem reasonable and help confirm that the dataset behaves realistically. Overall, the data looks clean and informative, with a few outliers and an uneven target distribution that we will handle in data preparation if needed.

- **Verifying data quality**

We checked the dataset to make sure it is suitable for modelling. All columns contain clean numeric values, and the format is the same across the whole file. There are no missing values, and the structure of the dataset is easy to understand. Most features stay within normal ranges for red wine, and the values make sense when looking at them together. The only exception is two very high values in “total sulfur dioxide”, which are much higher than usual. These are strong outliers and may need special handling later, but they do not make the dataset unusable.

Overall, the data quality is good. The dataset is complete, consistently formatted and does not contain errors that would stop us from building models. Any remaining issues, such as very high sulfur dioxide values or different feature scales, will be handled in the data preparation phase.

Planning your project

Tasks:

1. Finding the dataset to use (Tanel: 5h, Marten: 5h, Sven: 5h) - This takes into account the time we wasted because of the made up data of the Water Quality dataset -
Tools: Kaggle and the internet
2. Create the report (Tanel: 6h, Marten: 6h, Sven: 6h)
Tools: Jupyter Notebook, Google Docs
3. Prepare the data and train the SVM RBF model on the data (Tanel: 5h)
Tools: Jupyter Notebook, scikit-learn (sklearn), SVC (classification) and or SVR (regression)
4. Prepare the data and train the XGBoost model on the data (Sven: 5h)
Tools: Jupyter Notebook, scikit-learn (sklearn), XGBoost, XGBClassifier and or XGBRegressor
5. Prepare the data and train the Random Forest Classifier model on the data (Marten: 5h)
Tools: Jupyter Notebook, scikit-learn (sklearn), RandomForestClassifier
6. Assessing and comparing model performance (Tanel: 2h, Marten: 2h, Sven: 2h)
Tools: Jupyter Notebook, roc_auc_score
7. Create code and plots to analyze feature importance and correlation between quality or each other (Tanel: 4h, Marten: 4h, Sven: 4h)
Tools: Jupyter Notebook, Matplotlib, seaborn
8. Documenting and commenting (Tanel: 2h, Marten: 2h, Sven: 2h)
Tools: Jupyter Notebook, GitHub
9. Designing the poster (Tanel: 4h, Marten: 4h, Sven: 4h)
Tools: Canva
10. Presenting the poster (Tanel: 4h, Marten: 4h, Sven: 4h) - This includes presenting (3 hours) + and preparations (1 hour)
Tools: Tanel, Marten, Sven