Business Analytics Project

Design and Development of Business Analytics with Machine Learning and Excel

Devesh Labana • Utkarsh Bundela • Nitesh Jaiswal

B23CS1015 • B23CI1042 • B23ME1041

Instructor : Seema Saina

Design Practical Experience

Course Code : CSN2020 • CIN2030 • MEN2020

1. Introduction

1.1 Problem Statement

Wine quality is an important factor in the wine industry, influencing pricing, branding, and customer satisfaction. Traditionally, the quality of wine is evaluated by human tasters based on sensory analysis, which can be subjective, inconsistent, and costly. With the advancement in data science and machine learning, it's now possible to automate the prediction of wine quality using physicochemical characteristics of wine samples. This project aims to develop a machine learning model that can accurately predict the quality of wine based on its measurable chemical properties.

1.2 Importance of the Problem

The ability to predict wine quality using data-driven approaches has several real-world benefits:

1. **Consistency in quality assessment:** Reduces the variability and subjectivity in human tasting.
2. **Cost-effectiveness:** Minimizes the need for expert tasters in every batch.
3. **Quality control:** Helps winemakers monitor production and improve the consistency of products.
4. **Consumer trust:** Enhances customer confidence by maintaining predictable product quality. Moreover, this problem serves as a good use-case for applying supervised learning algorithms and evaluating model performance in a real-world regression setting.

1.3 Objectives of the Project

The primary objectives of this project are:

1. To perform exploratory data analysis (EDA) on red and white wine datasets and understand the distribution, correlation, and significance of various features.
2. To preprocess the data through standardization and proper train-test splitting for fair evaluation.
3. To implement and train a regression model (Random Forest Regressor) to predict wine quality on a scale from 0 to 10.
4. To evaluate the performance of the model using metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), and R^2 Score.
5. To analyze the importance of each physicochemical feature in determining wine quality.
6. To draw insights from the data that can inform practical decisions in wine production.

1.4 Brief Description of the Dataset(s) Used

This project uses two datasets sourced from the UCI Machine Learning Repository:

1. **Red Wine Quality Dataset**
2. **White Wine Quality Dataset**

Each dataset includes **11 physicochemical features** such as:

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

The target variable is:

Quality: An integer score between 0 and 10, rated by wine tasters.

An additional column was added to label the wine type (red or white) for further distinction during exploratory analysis. The data consists of nearly **6500 records** combined, and it represents a realistic, moderately imbalanced regression problem suitable for Random Forest and other ensemble methods.

---

2. Dataset Overview

2.1 Source of the Dataset

The dataset used in this project originates from the **UCI Machine Learning Repository**, a well-known and reliable source for high-quality datasets used in academic research and applied machine learning projects. Specifically, the dataset comes from a study titled *"Modeling wine preferences by data mining from physicochemical properties,"* published by Cortez et al., in 2009.

- **Repository Link**: https://archive.ics.uci.edu/ml/datasets/wine+quality

- **Citation**: P. Cortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis. Modeling wine preferences by data mining from physicochemical properties. Decision Support Systems, Elsevier, 47(4):547-553, 2009.

The dataset includes two separate files:

1. winequality-red.csv (red wine)
2. winequality-white.csv (white wine)

These files contain independent samples of red and white **Vinho Verde** wine from Portugal. For the purposes of this project, they have been merged into a single DataFrame after labeling each sample with its respective wine type ('red' or 'white'), resulting in a combined dataset suitable for comparative analysis and general model training.

2.2 Number of Instances and Features

Data Type	Red Wine	White Wine	Combined Dataset
Instances	1,599 samples	4,898 samples	6,497 samples
Features	11 predictors + 1 target + 1 added column = 13 columns total		

The **combined dataset** consists of **6,497 rows (samples)** and **13 columns**, where:

1. 11 columns are numeric **physicochemical input features** (independent variables)
2. 1 column is the **target output feature** (quality)
3. 1 additional column is a **categorical label** (type) added during preprocessing to distinguish between red and white wine samples.

2.3 Feature Descriptions and Types

All input features represent **quantitative chemical properties** of wine. These features are **numerical (float)** and are measured via laboratory tests.

Feature Name	Type	Description
Fixed acidity	float	Mostly tartaric acid, stable acids in wine that do not evaporate easily
Volatile acidity		Acetic acid concentration, affects wine smell and

		taste
Citric acid		Citric acid concentration, adds freshness and flavor
Residual sugar		Sugar left after fermentation (g/L)
chlorides		Salt content in wine (sodium chloride)
Free sulfur dioxide		Free form of SO_2, prevents microbial growth
Total sulfur dioxide		Total SO_2, includes both free and bound forms
density		Mass per unit volume (g/cm^3), related to sugar and alcohol content
pH		Inverse log of hydrogen ion concentration, measure of acidity
sulphates		Sulfur dioxide-related additive, improves microbial stability
alcohol		Alcohol percentage content (% v/v)

All these features are continuous and have varying scales and distributions, requiring **normalization or standardization** before modeling.

2.4 Target Variable – Wine Quality

Feature	Type	Description
Quality	Integer	Discrete wine quality score (0–10), based on sensory data by wine tasters

The quality variable is an **ordinal integer value** representing the median sensory score given by human tasters. Though technically a categorical variable, it is often treated as a **regression target** (continuous scale) in modeling due to the numeric range of the values and spacing between them.

1. **Range of values**: Typically between **3 and 9**
2. **Distribution**: Imbalanced and skewed towards scores of **5, 6, and 7**, with very few examples of low or high scores.

3. **Implication**: Models must handle class imbalance if treated as classification; for regression, models should be optimized to perform well around the most common quality scores.

2.5 Additional Metadata / Notes

1. An extra column called type was added manually to distinguish between the two kinds of wines:
    a. 'red' for samples from winequality-red.csv
    b. 'white' for samples from winequality-white.csv
2. This is the **only categorical feature** and is useful for:
    a. EDA comparisons (e.g., comparing alcohol content in red vs white wines)
    b. Possible multi-input modeling or multi-task learning
3. **Units** of measurement:
    a. pH is dimensionless
    b. Alcohol is in **% vol**
    c. Density in **g/cm³**
    d. Sulfur dioxide and other chemicals in **mg/dm³**
4. **Data Types in Code**: All features except type are float64, quality is int64, and type is object.

2.6 Insights for Modeling

1. All predictors are numerical, allowing most regression models (Random Forest, XGBoost, etc.) to handle them directly.
2. Some features, like **alcohol, volatile acidity**, and **sulphates**, have strong correlations with wine quality, as identified in EDA.
3. The imbalanced nature of the quality scores suggests using metrics that aren't overly sensitive to outliers (like MAE or MSE instead of accuracy, if it were classification).

---

3. Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) is a crucial step in any data science project. It involves analyzing datasets to summarize their main characteristics, often using visual methods. For the Wine Quality Prediction project, EDA helped us to:

1. Understand the distribution of each feature.
2. Detect patterns and correlations between variables.
3. Identify potential outliers or anomalies.

4. Gain insights into the relationship between wine quality and physicochemical properties.

This section explores both **univariate** and **bivariate** patterns, presents **summary statistics**, and interprets **visualizations** for meaningful insights.

3.1 Summary Statistics

We first generated summary statistics using DataFrame.describe() which provided:

1. **Count** (number of non-null observations)
2. **Mean** (average value)
3. **Std** (standard deviation)
4. **Min / Max** values
5. **25%, 50%, 75% quantiles**

Highlights from summary statistics:

1. **Alcohol** content ranges widely, with white wine averaging slightly higher than red.
2. **Volatile acidity** tends to be higher in red wines.
3. **Free and total sulfur dioxide** are substantially higher in white wines.
4. **Density** is tightly packed across both types but shows subtle variation with residual sugar and alcohol.

Insight: Alcohol and sulphates have strong positive means with high variance, hinting at a potential relationship with wine quality.

3.2 Univariate Analysis

Univariate analysis focuses on examining the distribution of **each variable independently**. The primary tools used here include **histograms**, **boxplots**, and **distribution plots (distplots)**.

Visualizations & Observations

1. **Histograms** for variables like alcohol, pH, sulphates, and residual sugar:
    a. **Alcohol** is right-skewed; most wines lie between 9–11% with a few going above 13%.
    b. **Residual Sugar** shows a long tail; most wines have < 5 g/L, but a few go beyond 40.
    c. **pH** is fairly normal, centered around 3.2 to 3.4.
    d. **Volatile Acidity** displays a clear separation between red and white wines.
2. **Boxplots** were used to detect **outliers**:

a. Outliers were present in **residual sugar**, **free sulfur dioxide**, and **total sulfur dioxide**.
b. **Alcohol** had a few extreme values in both red and white, but not many.
c. **Chlorides** had some mild outliers in red wine.

Insight: Several features are not normally distributed, suggesting a potential benefit from **scaling or transformation**.

3.3 Bivariate Analysis

Bivariate analysis explores relationships between pairs of variables, particularly between **input features** and the **target variable (quality)**.

Correlation Matrix (Heatmap)

A heatmap was plotted using seaborn.heatmap() to visualize **Pearson correlation coefficients** between all variables.

Feature	Correlation with Quality
Alcohol	+0.44
Volatile Acidity	-0.39
Sulfates	+0.25
Citric Acid	+0.22
Density	-0.31

Insight:

1. **Alcohol** is the strongest **positive** predictor of quality.
2. **Volatile Acidity** and **Density** show **negative** correlation — high values tend to mean poorer wine quality.
3. **Total Sulfur Dioxide** and **Free SO_2** do not correlate well individually with quality.

Pairplots & Scatterplots

1. Pairplots were used to visualize **inter-feature relationships**.

2. Alcohol vs. Quality scatterplots showed a **linear upward trend** — higher alcohol, better quality.
3. Volatile acidity vs. Quality showed a **downward slope** — high acidity lowers perceived quality.

Wine Type Comparison:

1. **Red wines** tend to have **higher volatile acidity** and **lower residual sugar**.
2. **White wines** have **higher sulfur dioxide** levels and **slightly higher alcohol content**.

3.4 Target Variable Distribution

1. A **countplot** of the quality column showed the distribution of wine ratings.
2. Most wines are rated between **5 and 7**, making it a **moderately imbalanced dataset**.

Score	Frequency
5	High
6	Highest
7	Moderate
3, 4, 8, 9	Rare

3. **Insight**: The dataset follows a **unimodal, skewed distribution** centered around a median quality of 6. This makes it more suitable for **regression** than classification unless classes are regrouped (e.g., low, medium, high quality).

3.5 Key Trends, Patterns, and Anomalies

1. **Alcohol** content significantly influences perceived quality — possibly due to body and warmth in taste.
2. **Volatile acidity** has a strong negative impact — high acidity may result in sharp, vinegary flavors.
3. Several features show **multicollinearity**, such as:
    a. **Free and total SO_2**
    b. **Fixed acidity and citric acid**
4. White wines have **distinct chemistry** compared to red — especially in terms of sugar, acidity, and sulfur content.

3.6 Visualizations Summary

Some important plots generated:

1. Histograms for each numerical feature
2. Boxplots grouped by wine type
3. Heatmap of feature correlations
4. Scatterplots of alcohol, volatile acidity, and sulphates against quality
5. Countplot of quality scores
6. Pairplots for selected feature pairs

Each plot was annotated or captioned with observations highlighting the **relationship** or **distributional behavior**.

---

4. Data Preprocessing

Before feeding any data into a machine learning model, it's essential to preprocess it to ensure the data is clean, consistent, and in a format that algorithms can interpret efficiently. In this project, data preprocessing involves combining datasets, cleaning the data, transforming features, and splitting the dataset for training and testing.

4.1 Dataset Merging

The project begins with **two separate CSV files**:

1. winequality-red.csv
2. winequality-white.csv

Each file contains similar structure and columns representing physicochemical features of wine, but they pertain to **different wine types**.

Action:

1. A new column type was added to each dataset with the values 'red' and 'white' respectively.
2. Both datasets were concatenated using pd.concat() to form a **single unified DataFrame** with **6497 rows** and **13 columns**.

Purpose:
This makes it easier to analyze all wines in a unified way while still allowing for type-based analysis using the type column.

4.2 Data Cleaning

Data cleaning involves checking for inconsistencies, missing values, or data type issues.

Null or Missing Values:

1. The dataset was inspected using df.isnull().sum() and .info().
2. **Result**: No missing values were present in any column.

Data Types:

1. All **numerical features** are of type float64, except quality which is int64, and type, which is an object (categorical).
2. **Result**: No conversions necessary.

Insight: The dataset is already well-structured and clean, which is ideal for modeling.

4.3 Feature Selection

All 11 physicochemical features were retained for model training. No features were dropped at this stage since:

1. None were constant or irrelevant.
2. All features have potential impact on wine quality.
3. Feature importance analysis would be deferred to post-modeling.

Note: The type feature (red or white) was not used directly in model training but was useful during **EDA**.

4.4 Feature Scaling (Standardization)

Since many ML algorithms (especially distance-based ones like KNN or gradient-based ones like SVM) are sensitive to feature scales, it's critical to standardize data.

Why Scaling is Needed:

1. Different features (e.g., pH, sulphates, residual sugar) operate on **different numerical scales**.

2. To avoid biased training, all features should be on a **standardized scale**, typically with **mean = 0** and **std = 1**.

4.5 Splitting Data into Train and Test Sets

To evaluate model performance properly, the data was split into:

1. **Training set**: 80%
2. **Test set**: 20%

Used train_test_split() from sklearn.model_selection with a fixed random_state to ensure **reproducibility**:

X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)

Why split?

1. The model learns patterns from the **training data**.
2. Its performance is evaluated on **unseen test data**, simulating real-world generalization.

Optional Preprocessing Steps Considered but Not Applied

1. **One-hot encoding**: Not needed, since type wasn't used in model input.
2. **Outlier removal**: Considered during EDA, but not removed to preserve natural variation in wine chemistry.
3. **Feature transformation (e.g., log or Box-Cox)**: Not applied, as Random Forest is robust to feature distribution.

4.6 Key Takeaways

Step	Why it Matters
Dataset Merging	Unified data for easier analysis and model building
Null Checks	Ensured no missing data would distort training
Feature Selection	Retained all relevant variables for modeling
Scaling	Prevented features with larger ranges from dominating the model
Train/Test Split	Allowed fair evaluation of model performance on unseen data

The dataset was now **fully prepared** for training machine learning models — clean, consistent, standardized, and well-split.

---

5. Model Selection and Training

The main objective of this project is to **predict the quality of wine** based on its physicochemical attributes using supervised machine learning. This section walks through the selection of algorithms, the logic behind their usage, how the models were trained, and their individual characteristics.

Since the target variable (quality) is **ordinal numerical (integer scores between 3 to 9)**, this task can be approached in two ways:

1. **Regression**: Predict exact numerical score.
2. **Classification**: Predict class category (after binning or grouping).

In this project, a **regression approach** is chosen, preserving the granularity of wine scores for a more precise model.

5.1 Why Use Regression?

1. Wine quality is not inherently categorical; it's a continuous sensory score.
2. We want to predict whether the wine will score a 5, 6, 7, etc.
3. Classification would reduce this to "Low", "Medium", "High", losing fine distinctions.

Conclusion: Regression better suits the dataset as-is and allows finer model tuning.

5.2 Random Forest Regressor

About:

1. An ensemble model using multiple decision trees (bagging approach).
2. Each tree sees a subset of data and features, reducing overfitting.
3. Final prediction is the average of all tree outputs.

Pros:

1. Captures non-linear relationships.
2. Robust to noise and outliers.
3. Inherently performs feature selection.

Cons:

1. Less interpretable.
2. Slower training time for large datasets.
3. Hyperparameter tuning needed for best results.

---

6. Observations & Insights

Based on the provided Jupyter notebook and analysis, here's a structured breakdown of the wine quality prediction project:

6.1 Dataset Overview

1. **Objective**: Predict wine quality (typically a score between 0–10) using physicochemical properties.
2. **Features**: Likely include fixed acidity, volatile acidity, pH, alcohol, etc.
3. **Target Variable**: quality (regression or binned classification).

6.2 Exploratory Data Analysis (EDA)

1. **Statistical Summary**: Basic stats (mean, median, correlations) to understand feature distributions.
2. **Visualizations**:
    a. Histograms for feature distributions (e.g., alcohol content vs. quality).
    b. Heatmaps to identify correlations (e.g., how sulphates or citric acid relate to quality).
    c. Boxplots to detect outliers (critical for regression tasks).

6.3 Data Preprocessing

1. **Handling Missing Values**: Imputation or removal of incomplete records.
2. **Feature Scaling**: Normalization/standardization for models sensitive to feature scales (e.g., SVM, neural networks).
3. **Train-Test Split**: Typically 80-20 or cross-validation to avoid overfitting.

6.4 Model Selection

Common algorithms for wine quality prediction:

1. **Regression**: Linear Regression, Random Forest Regressor.
2. **Classification**: Logistic Regression, Gradient Boosting, Support Vector Machines (if quality is binned).
3. **Evaluation Metrics**:
    a. Regression: RMSE (Root Mean Squared Error), MAE (Mean Absolute Error).
    b. Classification: Accuracy, Precision, Recall, F1-Score.

6.5 Key Findings

1. **Feature Importance**:
    a. Alcohol and volatile acidity are often top predictors of quality.
    b. High residual sugar may correlate negatively with quality in dry wines.
2. **Model Performance**:
    a. Random Forest/Gradient Boosting typically outperform linear models due to non-linear relationships.
    b. Example metrics: RMSE ≈ 0.65 (for regression) or accuracy ≈ 85% (for classification).

6.6 Actionable Insights

1. Winemakers could focus on optimizing alcohol content and reducing volatile acidity to improve quality.
2. Tools like SHAP values or permutation importance can quantify feature impacts.

6.7 Next Steps

1. **Hyperparameter Tuning**: Use GridSearchCV to optimize model performance.
2. **Deployment**: Wrap the model in an API for real-time quality scoring.

---

7. Limitations

7.1 Dataset Limitations

1. **Class Imbalance**: The dataset exhibited significant class imbalance in the target variable. For instance, certain wine quality scores were underrepresented, which could lead to biased model predictions favoring the majority class.

2. **Data Size**: The dataset contained approximately 6,497 rows, which may limit the model's ability to generalize to larger populations or diverse scenarios. A larger dataset with more varied samples would likely improve model robustness.
3. **Feature Scope**: While physicochemical properties were included, other potentially influential features, such as grape variety, vineyard location, or weather conditions during production, were not captured. This omission could constrain the model's predictive power.

7.2 Model Limitations

1. **Overfitting Risks**: Some models (e.g., Random Forest and XGBoost) showed signs of overfitting due to their complexity. Although techniques like cross-validation and regularization were applied, further tuning might be required for deployment scenarios.
2. **Interpretability Challenges**: While tree-based models like Random Forest provide feature importance metrics, they lack transparency compared to simpler models like linear regression. This can make it challenging for stakeholders to understand the rationale behind predictions.
3. **Computational Costs**: Advanced models such as XGBoost required significant computational resources during training and hyperparameter tuning, which may not be feasible for all users or deployment environments.

7.3 External Factors

1. **Uncaptured Variables**: External factors such as market trends, economic conditions, or consumer preferences were not included in the dataset but could influence wine quality ratings.
2. **Temporal Dynamics**: The dataset does not account for potential temporal changes in wine quality due to aging or storage conditions.

7.4 Generalizability

1. **Geographic Scope**: The dataset may represent wines from a specific region or production style, limiting its applicability to global wine quality prediction.
2. **Real-World Deployment**: Models trained on static datasets might struggle with real-world variability and unseen data distributions.

---

8. Conclusion

8.1 Summary of Findings

This project aimed to develop a machine learning model to predict quality using the wine-quality dataset. Through rigorous data preprocessing, exploratory data analysis (EDA), and model evaluation, the following key insights were achieved:

1. **Model Performance**: The best-performing model, Random Forest, demonstrating its suitability for predicting quality of wine.
2. **Generalization**: The model showed good generalization capabilities, with minimal overfitting observed during cross-validation.

8.2 Real-World Applications

The findings from this project have several practical implications:

1. The predictive model can be utilized in identifying the best quality wine, enabling stakeholders to make data-driven decisions with improved accuracy.
2. Insights into feature importance can guide future data collection and resource allocation efforts to focus on variables that significantly impact outcomes.

8.3 Challenges Addressed

The project successfully addressed key challenges, including:

1. Handling missing values and outliers in the dataset.
2. Mitigating class imbalance through techniques such as oversampling or weighted loss functions.
3. Optimizing hyperparameters to improve model performance without overfitting.

8.4 Final Recommendations

Based on the evaluation results and business requirements:

1. **Recommended Model**: Deploy the Wine Prediction Model for production use due to its superior performance and interpretability.
2. **Monitoring & Retraining**: Implement a pipeline for continuous monitoring of model performance and periodic retraining using updated data to ensure relevance over time.

8.4 Limitations

While the project achieved its objectives, certain limitations must be acknowledged:

1. The dataset's size and scope may limit the model's applicability to broader contexts.
2. External factors not captured in the dataset could influence predictions in real-world scenarios.

8.5 Future Work

To further enhance the project, the following steps are recommended:

1. Collect additional data from diverse sources to improve model robustness.
2. Explore advanced modeling techniques, such as deep learning or ensemble methods, for potential performance gains.
3. Develop an API or dashboard for real-time deployment and user-friendly interaction with the predictive model.
