Quality of Wine Analysis Report

Math 4322: Introduction to Data Science and Machine Learning

Dr. Cathy Poliak

Group10: Arjun Karnani, Dhanush Kumar Selvaraj Kumar, Naumi Aparanji,

Saptarshi De, Shruthi Yenamagandla

November 26, 2024

Table of Contents

Introduction 3

- Dataset Description 3
- Inputs and Outputs 3
- Research Question 3
- Importance of the Question 4
- Scope of Analysis 4

Methods 5

- Linear Regression Model 5
  - Why we chose this model 5
  - Formula 5
  - Base Model 6
  - Pruned Model 8
  - Results 10
- Tree-Based Models 11
  - Why we chose this model 11
  - Formula 11
  - Split into training and testing 11
  - Basic Tree 12
  - Pruned Tree 14
  - Bagged Tree 16
  - Random Forests 19
  - Results 21

Conclusion 22

Bibliography 22

Introduction

Dataset Description

The Red Wine Quality dataset is a collection of data points representing the physicochemical properties of red wine samples produced in the Vinho Verde region of Portugal. The dataset consists of 11 input variables or features that describe the chemical characteristics of wine and a single output variable (`quality`) that represents the quality rating assigned by wine tasters on a scale from 0 to 10.

Key features in the dataset include:

- Fixed Acidity: The concentration of non-volatile acids, contributing to the tartness of wine.
- Volatile Acidity: The amount of acetic acid, which can give wine a vinegar-like taste at high levels.
- Citric Acid: Adds freshness and enhances the flavor profile.
- Residual Sugar: The amount of sugar remaining after fermentation; higher levels can indicate sweetness.
- Chlorides: The salt content in the wine.
- Free Sulfur Dioxide: The free form of SO_2 in the wine, protecting it from microbial growth and oxidation.
- Total Sulfur Dioxide: Includes both bound and free forms of SO_2, influencing the preservative capacity.
- Density: Influences the body of the wine; closely related to sugar and alcohol content.
- pH: Measures the acidity or alkalinity of wine, affecting taste and microbial stability.
- Sulphates: Contributes to wine preservation and enhances the mouthfeel.
- Alcohol: A key determinant of wine quality, influencing the body and taste profile.

The target variable (`quality`) is an integer score based on sensory evaluations by wine experts, reflecting the overall perception of the wine.

Inputs and Outputs

- Inputs: Fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, and alcohol.
- Output: Quality rating (0 to 10), which serves as the response variable (`quality`).

Research Question

This project aims to answer the question:

"Can we predict the quality of red wine based on its physicochemical properties?"

We aim to build predictive models that can quantify the relationship between the input features and wine quality. By doing so, we hope to identify key physicochemical factors that influence wine quality and evaluate the MSE of different modeling approaches.

Importance of the Question

Understanding and predicting wine quality is valuable for winemakers and distributors to:

1. Optimize the production process by adjusting chemical properties.
2. Ensure consistent quality to meet consumer expectations.
3. Identify key factors that contribute to high-quality wine production.

Scope of Analysis

- Type of Analysis: Predictive modeling.
- Target Variable Type: Treated as continuous for regression models.
- Goals:
    1. Develop two machine learning models (Linear Regression and Random Forest).
    2. Compare their predictive performance to identify the most suitable approach for this dataset.
    3. Provide interpretability in terms of feature significance.

Methods

Linear Regression Model (Naumi Aparanji, Shruthi Yenamagandla)

Why we chose this model

We chose this model because linear regression is suitable when the relationship between the response variable (`quality`) and predictors is approximately linear, as suggested by the exploratory data analysis and diagnostic plots. It provides a clear and interpretable model to quantify how predictors influence wine quality.

Linear regression also provides statistical significance tests for each predictor (p-values) and the overall model (F-statistic), helping to identify the most important factors affecting wine quality.

Formula

$$\text{Quality} = \beta_0 + \beta_1 \cdot \text{fixed.acidity} + \beta_2 \cdot \text{volatile.acidity} + \beta_3 \cdot \text{citric.acid} + \beta_4 \cdot \text{residual.sugar} + \beta_5 \cdot \text{chlorides} + \beta_6 \cdot \text{free.sulfur.dioxide} + \beta_7 \cdot \text{total.sulfur.dioxide} + \beta_8 \cdot \text{density} + \beta_9 \cdot \text{pH} + \beta_{10} \cdot \text{sulphates} + \beta_{11} \cdot \text{alcohol} + \epsilon$$

Where:

- β_0 is the intercept.
- $\beta_1, \beta_2, \ldots, \beta_{11}$ are the coefficients for each predictor.
- ϵ represents the error term (residuals).

Base Model

We first trained our model with all the predictor variables in the dataset, and the response variable as `quality`. Here is the R code that we used to create our model:

```
> wine_model <- lm(quality ~ ., data = winedata)
> summary(wine_model)

Call:
lm(formula = quality ~ ., data = winedata)

Residuals:
     Min       1Q   Median       3Q      Max
-2.68911 -0.36652 -0.04699  0.45202  2.02498

Coefficients:
                       Estimate Std. Error t value Pr(>|t|)
(Intercept)           2.197e+01  2.119e+01   1.036   0.3002
fixed.acidity         2.499e-02  2.595e-02   0.963   0.3357
volatile.acidity     -1.084e+00  1.211e-01  -8.948  < 2e-16 ***
citric.acid          -1.826e-01  1.472e-01  -1.240   0.2150
residual.sugar        1.633e-02  1.500e-02   1.089   0.2765
chlorides            -1.874e+00  4.193e-01  -4.470 8.37e-06 ***
free.sulfur.dioxide   4.361e-03  2.171e-03   2.009   0.0447 *
total.sulfur.dioxide -3.265e-03  7.287e-04  -4.480 8.00e-06 ***
density              -1.788e+01  2.163e+01  -0.827   0.4086
pH                   -4.137e-01  1.916e-01  -2.159   0.0310 *
sulphates             9.163e-01  1.143e-01   8.014 2.13e-15 ***
alcohol               2.762e-01  2.648e-02  10.429  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.648 on 1587 degrees of freedom
Multiple R-squared:  0.3606,    Adjusted R-squared:  0.3561
F-statistic: 81.35 on 11 and 1587 DF,  p-value: < 2.2e-16
```

Based on the summary above we were able to understand the significance on each predictor in predicting the response variable. We classified significant predictors as the variables with p-values < 0.05.

Significant variables:
`volatile.acidity`, `chlorides`, `free.sulfur.dioxide`, `total.sulfur.dioxide`, `pH`, `sulphates`, and `alcohol`

Insignificant variables:
`fixed.acidity`, `citric.acid`, `residual.sugar`, and `density`

Adjusted R^2:
The adjusted R^2 value (0.3561) indicates that about 35.61% of the variance in quality is explained by the predictors. While not very high, this may be reasonable depending on the dataset.

Residual Standard Error (RSE): The RSE of 0.648 indicates the typical deviation of the observed `quality` values from the model's predictions.

Model Fit: The overall p-value (< 2.2e-16) indicates that the model as a whole is statistically significant.
We then proceeded to calculate the MSE for the base model. Here is the R code:

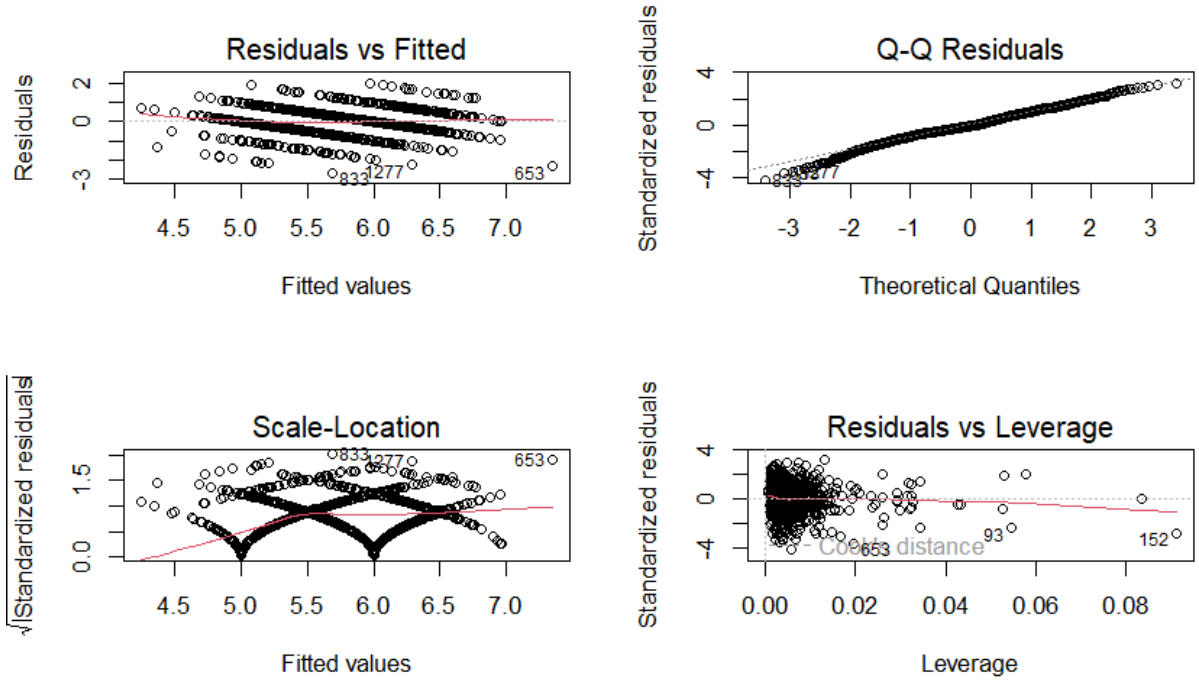
```
> # Calculate MSE for the base model
> residuals2 <- wine_model$residuals
> mse2 <- mean(residuals2^2)
> print(mse2)
[1] 0.4167672
```

Pruned Model

After creating the base model and understanding which predictors play a significant role, we pruned our model and trained the new one only on the significant predictors. Here is the R code:

```
> refined_model <- lm(quality ~ volatile.acidity + chlorides +
free.sulfur.dioxide + total.sulfur.dioxide + pH + sulphates +
alcohol, data = winedata)
> summary(refined_model)

Call:
lm(formula = quality ~ volatile.acidity + chlorides +
free.sulfur.dioxide +
    total.sulfur.dioxide + pH + sulphates + alcohol, data = winedata)

Residuals:
     Min       1Q   Median       3Q      Max
-2.68918 -0.36757 -0.04653  0.46081  2.02954

Coefficients:
                       Estimate Std. Error t value Pr(>|t|)
(Intercept)           4.4300987  0.4029168  10.995  < 2e-16 ***
volatile.acidity     -1.0127527  0.1008429 -10.043  < 2e-16 ***
chlorides            -2.0178138  0.3975417  -5.076 4.31e-07 ***
free.sulfur.dioxide   0.0050774  0.0021255   2.389    0.017 *
total.sulfur.dioxide -0.0034822  0.0006868  -5.070 4.43e-07 ***
pH                   -0.4826614  0.1175581  -4.106 4.23e-05 ***
sulphates             0.8826651  0.1099084   8.031 1.86e-15 ***
alcohol               0.2893028  0.0167958  17.225  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6477 on 1591 degrees of freedom
Multiple R-squared:  0.3595,    Adjusted R-squared:  0.3567
F-statistic: 127.6 on 7 and 1591 DF,  p-value: < 2.2e-16

> par(mfrow = c(2, 2))
> plot(refined_model)
```

We get the following plots:



Based on this summary and plots from the refined model, here are the inferences we made.

Significant Predictors:
All predictors in the refined model are statistically significant ($p<0.05p < 0.05p<0.05$), meaning they contribute meaningfully to predicting `quality`.
Strongest predictors (based on t-values): `alcohol`, `volatile.acidity`, `sulphates`.

Adjusted R^2:
Adjusted $R^2 = 0.3567$: About 35.67% of the variance in `quality` is explained by the model. While this is reasonable for real-world datasets, there's still room for improvement.

Residual Standard Error (RSE):
RSE =0.6477: Indicates the typical deviation of observed `quality` values from the predicted values. Lower RSE suggests better model fit.

F-statistic:
p-value <2.2×10^{-16}: The overall model is statistically significant.
Coefficients:
`volatile.acidity` and `chlorides` have negative effects on `quality`, meaning higher values for these variables tend to reduce wine quality.
`alcohol`, `sulphates`, and `free.sulfur.dioxide` positively influence `quality`.

Linearity (Residuals vs Fitted Plot):
The residuals are scattered around zero but show slight clustering and curvature, indicating mild non-linearity. Consider adding interaction or polynomial terms.

Normality (Q-Q Plot):
Deviations in the tails suggest slight non-normality in residuals. Outliers might be influencing this; consider robust regression or transformations.

Equal Variance (Scale-Location Plot):
Increasing spread of residuals at higher fitted values indicates heteroscedasticity. A log or Box-Cox transformation of the response variable may help.

Influential Points (Residuals vs Leverage Plot):
Influential wines (like those at points 152 and 653) could provide insights into unique characteristics or errors that need attention

We then proceeded to calculate the MSE for the pruned model. Here is the R code:

```
> # Calculate MSE for the refined model
> residuals <- refined_model$residuals
> mse <- mean(residuals^2)
> print(mse)
[1] 0.4174716
```

The MSE for the pruned model is very slightly higher than the base model. This could be because even though the predictors that we cut out in the pruned model had low significance values, they were still needed to accurately predict the `quality` of the dataset.

Results

Overall, the linear regression model (base model) produced an MSE of **0.4167.** MSE measures the average squared difference between the estimated values that our model predicts given certain parameters and the actual values observed in the test sample. A more accurate model will have an MSE closer to 0 and vice versa. By that assessment, we conclude that the quality of our model, wine_model, is fairly accurate in predicting the `quality`.

Tree-Based Models (Arjun Karnani, Dhanush Kumar Selvaraj Kumar, Saptarshi De)

Why we chose this model

For our second model, we decided to explore using a tree-based model. Specifically, we selected a regression decision tree since our response variable, wine quality, is a continuous numerical variable. A regression decision tree enables us to model the relationship between the physicochemical properties of red wine and its quality rating in an intuitive and interpretable manner. This approach allows us to visually represent the decisions that lead to the predicted quality scores based on the input variables.

Additionally, using a tree-based model provides flexibility to perform advanced techniques such as pruning, bagging, and random forests to refine and improve the model's performance. Pruning helps reduce the complexity of the tree, preventing overfitting and improving the generalization on new data. Bagging, by aggregating predictions from multiple decision trees, reduces variance and increases stability. Random forests, an extension of bagging, further enhance model accuracy by incorporating random feature selection at each split.

By using these techniques, we aim to optimize the predictive performance of the model while identifying key physicochemical properties that significantly impact wine quality.

Formula

```
quality ~ alcohol + sulphates + volatile.acidity +
total.sulfur.dioxide + fixed.acidity + citric.acid + residual.sugar +
chlorides + free.sulfur.dioxide + density + pH
```

Split data into training and testing

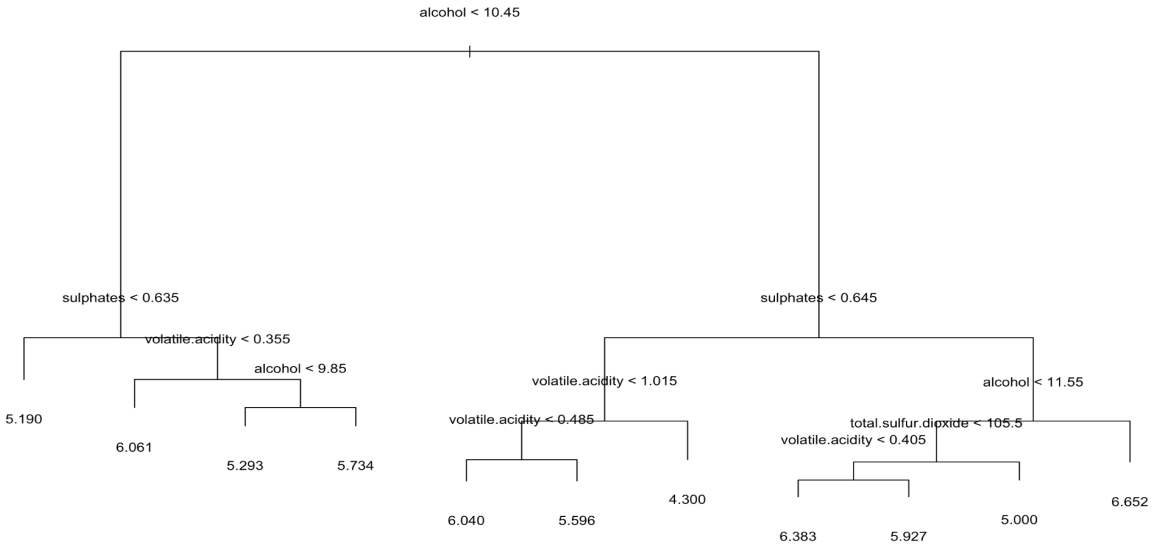
We decided to split our dataset as 80 percent into training and 20 percent into testing. This way, we have a good amount of data to train the model on but also have some data to test our trained model and make any improvements.

```
> set.seed(1)
> sample <- sample(nrow(dataset), nrow(dataset)*0.80)
> train <- dataset[sample,]
> test <- dataset[-sample,]
```

Basic Tree

To create a baseline for all of our decision tree-based models we created a base decision tree using tree() including all data used by our linear regression model.

```
> library(tree)
> tree <- tree(quality~., data = train)
> summary(tree)

Regression tree:
tree(formula = quality ~ ., data = train)
Variables actually used in tree construction:
[1] "alcohol"              "sulphates"            "volatile.acidity"
"total.sulfur.dioxide"
Number of terminal nodes:  11
Residual mean deviance:  0.3927 = 497.9 / 1268
Distribution of residuals:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-2.1900 -0.2933 -0.1903  0.0000  0.4043  1.9390

> plot(tree)
> text(tree, pretty = 1)
```



MSE

```
> prediction.original <- predict(tree, test)
> mse.original <- mean((prediction.original - test$quality)^2)
> mse.original
[1] 0.4801827
```

R^2

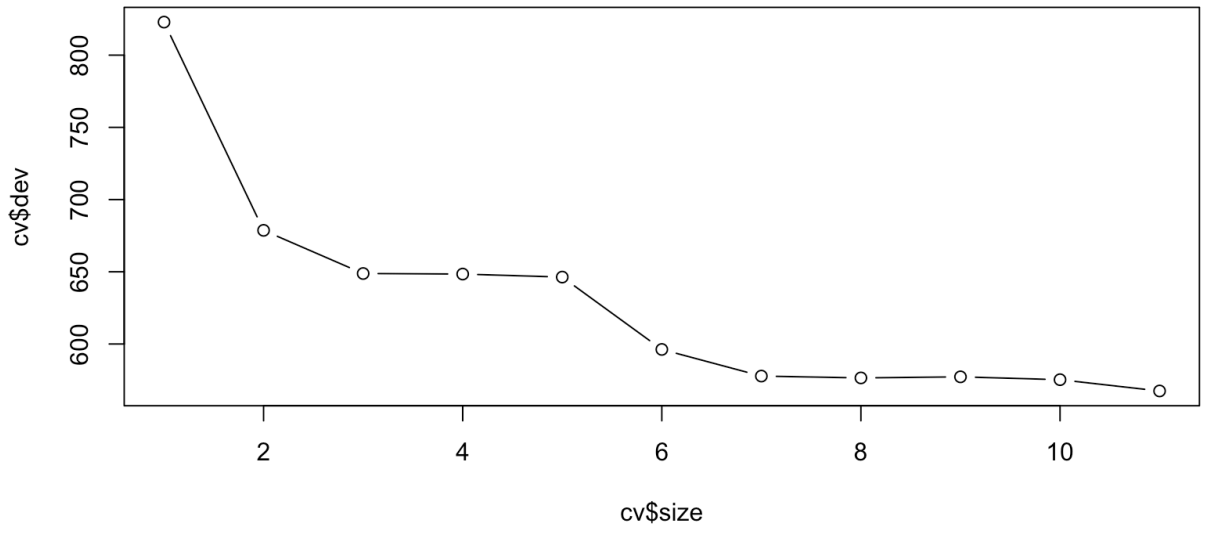
```
> prediction.original <- predict(tree, test)
> ssr <- sum((test$quality - prediction.original)^2)
> tss <- sum((test$quality - mean(test$quality))^2)
> r_squared <- 1 - (ssr / tss)
> r_squared
[1] 0.3031263
```

The residual mean deviance (0.3927) and **MSE** (0.4802) indicate the model's accuracy in capturing the variance in wine quality. However, the relatively low **R^2** (0.3031) suggests that a significant proportion of variance remains unexplained. This highlights the potential for improvement through techniques like pruning or incorporating additional variables.

Although the base decision tree provides a reasonable foundation, its complexity with 11 terminal nodes could lead to overfitting. By pruning the tree, we aim to simplify the model, reduce overfitting, and enhance its generalization to unseen data.

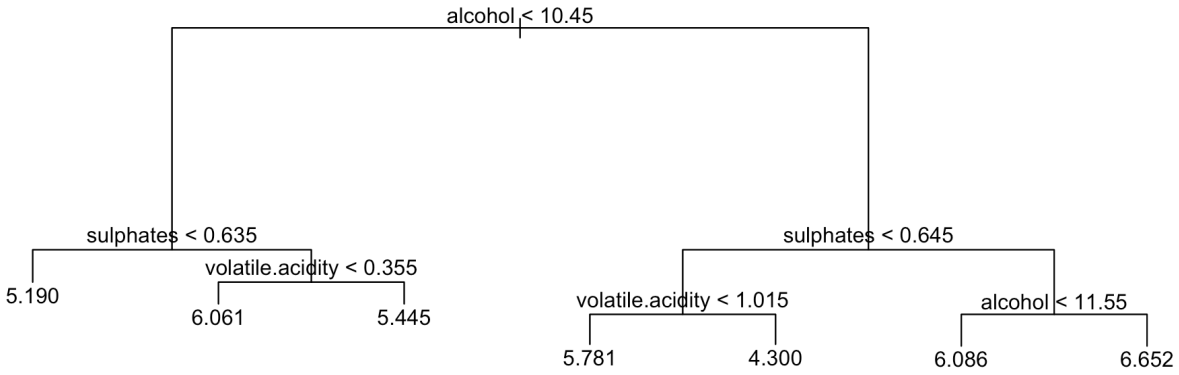
Pruned Tree

```
> cv <- cv.tree(tree, FUN=prune.tree)
> plot(cv$size, cv$dev, type = "b")
```



To simplify the model and potentially improve its generalization, we pruned the initial decision tree using cross-validation to determine the optimal complexity. Based on the cross-validation results, we reduced the tree to 7 terminal nodes:

```
> tree.pruned <- prune.tree(tree, best=7)
> summary(tree.pruned)

Regression tree:
snip.tree(tree = tree, nodes = c(14L, 11L, 12L))
Variables actually used in tree construction:
[1] "alcohol"          "sulphates"        "volatile.acidity"
Number of terminal nodes:  7
Residual mean deviance:  0.4234 = 538.5 / 1272
Distribution of residuals:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-2.1900 -0.4454 -0.1903  0.0000  0.5546  1.9390

> plot(tree.pruned)
> text(tree.pruned)
```



MSE

```
> prediction.pruned <- predict(tree.pruned, test)
> mse.pruned <- mean((prediction.pruned - test$quality)^2)
> mse.pruned
[1] 0.504505
```

R^2

```
> prediction.pruned <- predict(tree.pruned, test)
> ssr <- sum((test$quality - prediction.pruned)^2)
> tss <- sum((test$quality - mean(test$quality))^2)
> r_squared <- 1 - (ssr / tss)
> r_squared
[1] 0.2678281
```

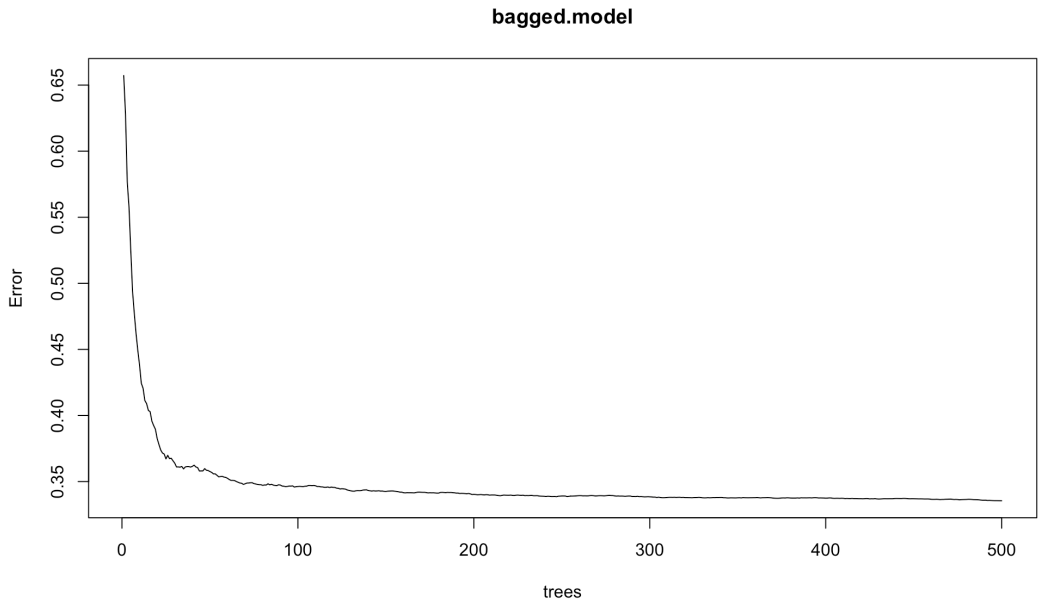
The pruned model exhibited a marginally higher **MSE** (0.5045) and a slightly lower **R^2** (0.2678) compared to the basic tree. These results suggest that while pruning simplified the model, it did not improve its predictive power. This trade-off is a typical outcome when balancing complexity and performance.

While the pruned model offers better interpretability and reduced complexity, its slight performance decline highlights the limitations of a single decision tree. To address this, we proceeded to create a bagged tree model, leveraging ensemble methods to enhance prediction accuracy and robustness.

Bagged Tree

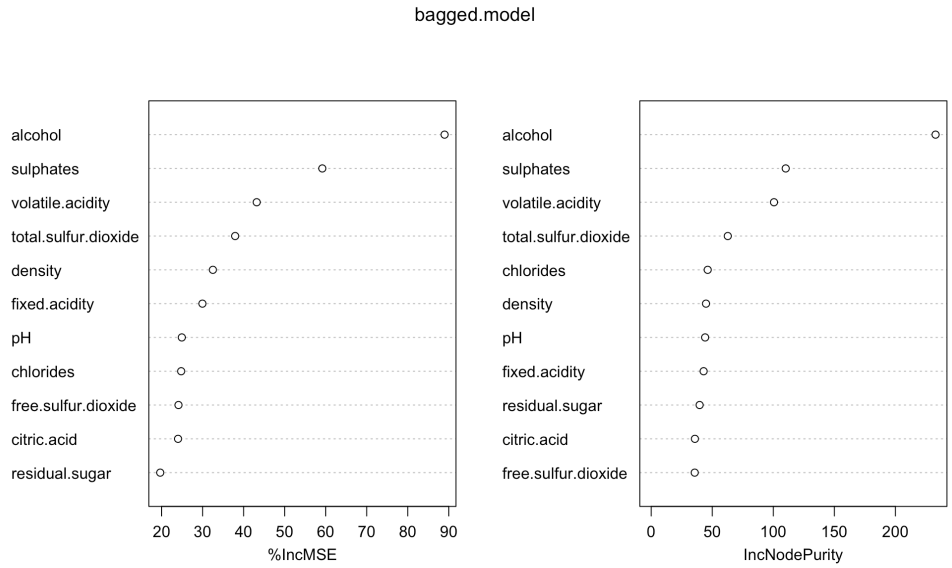
To enhance model performance and reduce variance, we implemented a bagged tree model using the `randomForest` package. Bagging creates multiple trees by bootstrapping the data and averages their predictions, offering greater stability compared to a single decision tree. Our bagged model used 500 trees and considered all predictors at each split:

```
> bagged.model <- randomForest(quality ~ ., data = train, mtry =
ncol(data) - 1, importance = TRUE)
> bagged.model

Call:
 randomForest(formula = quality ~ ., data = train, mtry = ncol(data)
-      1, importance = TRUE)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 11

          Mean of squared residuals: 0.3379373
                    % Var explained: 47.39

> plot(bagged.model)
```



The variable importance plot revealed that `alcohol` and `sulphates` were the most influential predictors, followed by `volatile acidity` and `total sulfur dioxide`. These results align with the earlier decision tree models but provide deeper insights into the relative contribution of each feature:

```
> varImpPlot(bagged.model)
```




```
> print(importance(bagged.model))
                          %IncMSE IncNodePurity
fixed.acidity           30.24319      42.38481
volatile.acidity        40.18690      98.78621
citric.acid             24.27844      36.49435
residual.sugar          20.23343      40.95123
chlorides               25.33328      47.85098
free.sulfur.dioxide     23.56585      35.40974
total.sulfur.dioxide    41.50003      62.81641
density                 30.08750      43.20483
pH                      24.51105      44.16791
sulphates               65.21349     109.13908
alcohol                 84.49842     233.40189
```

MSE

```
> prediction.bagged.model<- predict(bagged.model, test)
> mse.bagged.model <- mean((prediction.bagged.model - test$quality) ^
2)
> mse.bagged.model
[1] 0.3439731
```

R^2

```
> prediction.bagged.model <- predict(bagged.model, test)
> ssr <- sum((test$quality - prediction.bagged.model)^2)
> tss <- sum((test$quality - mean(test$quality))^2)
> r_squared <- 1 - (ssr / tss)
```

```
> r_squared
[1] 0.5008029
```

On the test set, the bagged model achieved an **MSE** of 0.344 and an **R^2** of 0.501, significantly outperforming both the basic and pruned trees. This demonstrates the effectiveness of ensemble methods in improving predictive accuracy and model robustness.

The bagged tree model provided robust performance, reducing error and explaining more variance compared to earlier models. Building on this success, we explored the behavior of the model using Random Forest methods to investigate whether additional improvements could be achieved through feature randomization.

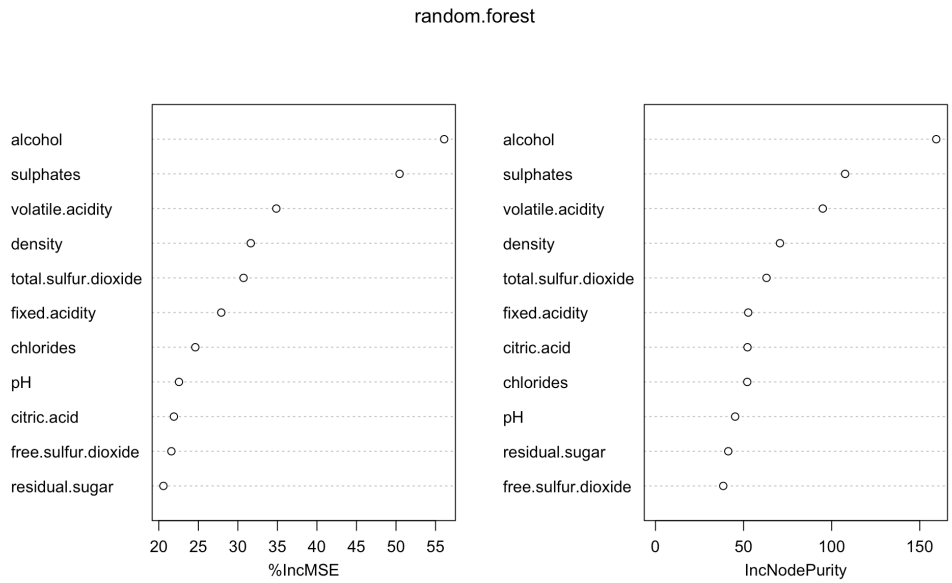
Random Forests

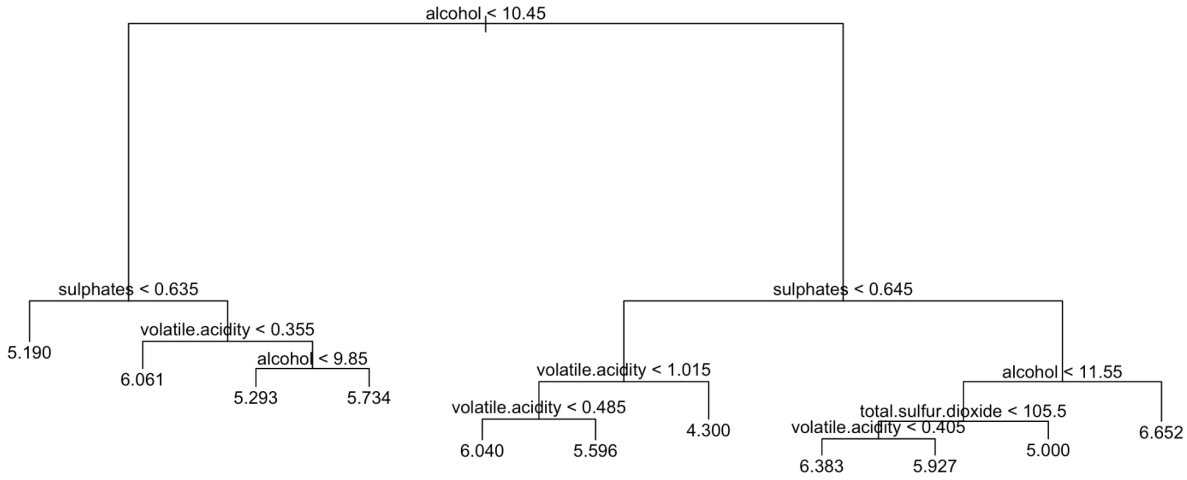
Building on the bagged tree approach, we developed a **Random Forest** model to further improve predictive performance. By allowing a random subset of variables at each split (mtry = 3), the Random Forest introduces additional variability, reducing overfitting and increasing accuracy. We used 500 trees in the ensemble:

```
> random.forest <- randomForest(quality ~ ., data = train, ntree =
500, mtry = 3, importance = TRUE )
> random.forest

Call:
 randomForest(formula = quality ~ ., data = train, ntree = 500,
mtry = 3, importance = TRUE)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 3

          Mean of squared residuals: 0.3304292
                    % Var explained: 48.56
```

The variable importance plot identified `alcohol` and `sulphates` as the most influential features, consistent with prior models. `Volatile acidity` and `density` also showed high importance, reaffirming their role in predicting wine quality.

```
> varImpPlot(random.forest)
```






```
> random.forest.tree <- tree(quality ~ ., data = train, subset =
random.forest$inbag)
> plot(random.forest.tree)
> text(random.forest.tree, pretty = 1)

> print(importance((random.forest)))
                      %IncMSE IncNodePurity
fixed.acidity        27.89774      52.63180
volatile.acidity     34.85799      94.94552
citric.acid          21.90637      52.21304
residual.sugar       20.58214      41.30400
chlorides            24.61561      52.08608
free.sulfur.dioxide  21.58890      38.45574
total.sulfur.dioxide 30.71550      63.00608
density              31.62826      70.64925
pH                   22.55077      45.21380
sulphates            50.44987     107.63633
alcohol              56.09139     159.35736
```

MSE

```
> prediction.random.forest <- predict(random.forest, test)
> mse.random.forest <- mean((prediction.random.forest - test$quality)
^ 2)
> mse.random.forest
[1] 0.3455529
```

R^2

```
> prediction.random.forest <- predict(random.forest, test)
> ssr <- sum((test$quality - prediction.random.forest)^2)
```

```
> tss <- sum((test$quality - mean(test$quality))^2)
> r_squared <- 1 - (ssr / tss)
> r_squared
[1] 0.4985103
```

The Random Forest model delivered the best overall performance, offering a balanced trade-off between accuracy, variance explained, and robustness. While the bagged tree was competitive, the additional randomization in Random Forest improved its generalization.

Results

Here are the results of all the models we used:

Model	MSE	R^2	% Variance Explained	Key Insights
Basic Tree	0.480	0.303	~30.3%	Simple, interpretable baseline model.
Pruned Tree	0.505	0.268	~26.8%	Simplified model with reduced complexity but lower accuracy.
Bagged Tree	0.344	0.501	~47.4%	Significant improvement in accuracy and variance explained.
Random Forest	0.346	0.499	~48.6%	Slightly better performance than bagged tree, highest variance explained.

Conclusion

This project involved evaluating several modeling techniques to predict wine quality based on its chemical properties. Starting with **linear regression**, we explored the significance of predictors and found that alcohol, sulphates, and volatile acidity were the most influential features. While the pruned regression model simplified interpretation, it showed minimal improvement in performance.

The **tree-based models** offered progressively better accuracy and robustness. The **bagged tree** and **random forest** models significantly outperformed linear regression and single decision trees, with the random forest slightly edging out in variance explained. These ensemble methods effectively captured interactions and reduced overfitting, highlighting their utility in complex datasets.

Overall, the **Random Forest model** emerged as the best-performing model, balancing accuracy, robustness, and interpretability. Future work could include exploring advanced ensemble techniques like gradient boosting, hyperparameter optimization, or experimenting with feature transformations to further refine model performance.

This project demonstrates the value of comparing multiple modeling approaches to uncover the most effective technique for a given dataset and problem.

Bibliography

Bansal, Lovish. Red Wine Quality Dataset. Kaggle, https://www.kaggle.com/datasets/lovishbansal123/red-wine-quality/data. Accessed 28 Oct. 2024.