Wine Quality Data Analysis by - Aadhar Agarwal

Table of Contents

Summary

This report presents the methodology and findings from the development and evaluation of predictive models for estimating wine quality, undertaken to demonstrate my analytical and machine learning competencies as a data analyst. The project involved an in-depth analysis of a dataset containing physicochemical tests of wines, with the goal of developing a reliable model to predict quality ratings. Among the models considered, a Linear Regression model employing backward selection was identified as the most suitable due to its balance of interpretability, simplicity, and accuracy. This model is recommended for its practicality and effectiveness, illustrating a successful application of data analysis in a real-world context.

Introduction

The wine industry values the objective assessment of wine quality due to its direct correlation with consumer preferences and market value. The goal of this project was to leverage available physicochemical data to predict wine quality scores, thus providing a tool that could potentially streamline quality assessment processes.

Data Overview and Preprocessing

Index	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	ae sulfur dioxic	tal sulfur dioxi	density	pH	sulphates	alcohol	quality
0	7.4	0.7	0	1.9	0.076	11	34	0.9978	3.51	0.56	9.4	5
1	7.8	0.88	0	2.6	0.098	25	67	0.9968	3.2	0.68	9.8	5
2	7.8	0.76	0.04	2.3	0.092	15	54	0.997	3.26	0.65	9.8	5
3	11.2	0.28	0.56	1.9	0.075	17	60	0.998	3.16	0.58	9.8	6

Figure 1. DataFrame

- *Figure 1* shows the set dataset used consists of physicochemical properties of wine as the input variables and a quality score as the output variable.
- The preprocessing steps included handling missing values, encoding categorical variables (if any), and removing near-zero variance features to reduce dimensionality and potential noise.

Exploratory Data Analysis (EDA)

EDA began with visualizations to understand data distributions and inter-variable relationships, followed by a correlation analysis to identify multicollinearity.

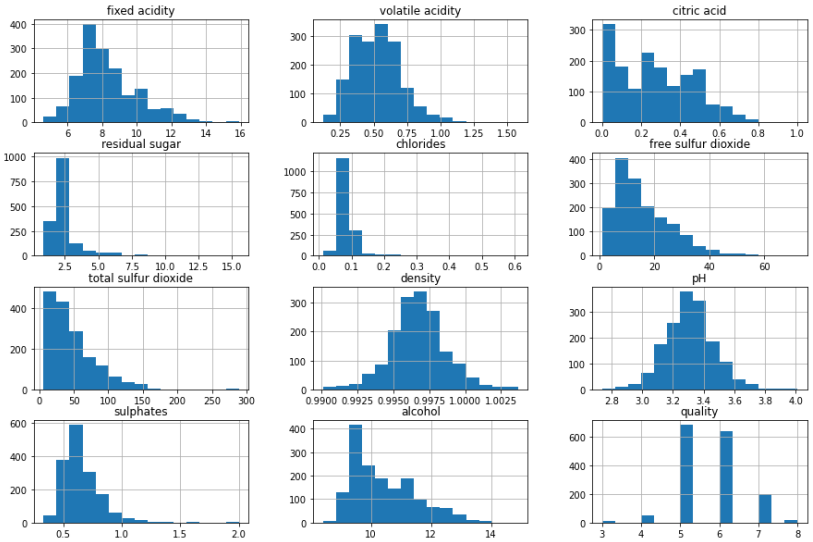

Figure 2. Data Distribution Histograms

- **Histograms** were generated to understand the distribution of each feature and the target variable, quality. The histograms would help identify skewness, outliers, and the need for data normalization or transformation as shown in *Figure 2*. One of the many insights from the histogram show that most of the wines have a quality level of 5 and 6 while there are no wines with quality below 3 and above 8.

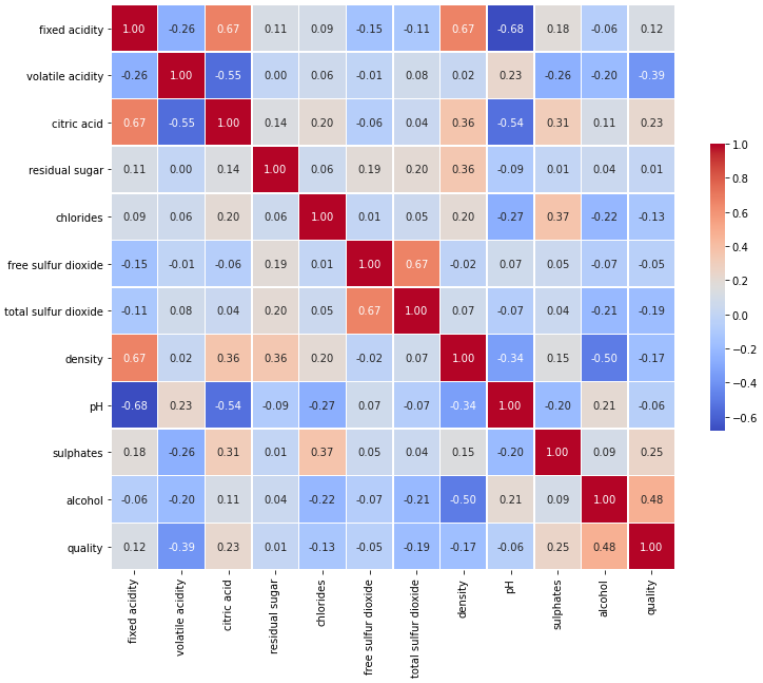

Figure 3. Correlation Matrix heat map

- A **Correlation Matrix** was created to identify the relationships between features, and with the target variable. It showed varying degrees of positive and negative correlations. *Figure 3*. shows that quality is positively correlated to alcohol and sulphate while volatile acidity is totally opposite.

Feature Selection and Model Building

Various models were tested, including **Linear Regression**, **Decision Trees**, **KNN**, and **Random Forest**. Each model brought its assumptions and capabilities to capture the underlying patterns in the data.

- **Relevance**: Predictors with near-zero variance were removed to improve the model's predictive accuracy. The exclusion of these predictors simplifies the model, enhancing interpretability and reducing the likelihood of overfitting.
- **Backward Selection**: This method was used to iteratively remove the least significant features based on their p-values. The aim was to simplify the model by keeping only the features with a significant impact on the target variable.
- **Cross-Validation**: Cross-validation was employed to assess the model's predictive performance and robustness rigorously. This technique is essential for identifying a model that not only fits the training data well but also performs consistently on unseen data.

Model Evaluation

Model Approach	Train MAE	Test MAE	Difference	RSME	MSE
Linear Model - All predictors	0.5	0.504	**0.004**	**0.635**	**0.39**
Linear Model - Remove correlated predictors, near zero variance	0.503	0.509	**0.006**	**0.63**	**0.396**
Linear Model - backward selection	0.5	0.504	**0.004**	**0.626**	**0.391**
Linear Model - CV with backward selected data	0.508	0.504	**-0.004**	**0.656**	**0.391**
Tree - CV (tuned on complexity)	0.504	0.541	**0.037**	**0.682**	**0.465**
KNN	0.449	0.579	**0.13**	**0.729**	**0.532**
RF - with backward selected data w/o CV	0.155	0.424	**0.269**	**0.307**	**0.554**
RF - All predictors w/o CV	0.154	0.422	**0.268**	**0.549**	**0.301**
RF - All predictors with CV	0.5	0.422	**-0.078**	**0.648**	**0.422**

Figure 4. Model Performance Table

Models were evaluated based on Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE). A summary table was created in MS Excel to keep track of each model's performance as shown in *Figure 4*. The primary observations were:

- **Linear Models** showed consistent performance on both training and test sets, with a very small difference in MAE, indicating good generalization.
- **Decision Trees and Random Forests** typically had lower MAE values, suggesting better prediction accuracy. However, the difference in MAE was greater while having a lower RMSE.
- **KNN** performed less well, potentially due to the scale-sensitive nature of the algorithm, pointing to the need for normalization or feature scaling.

The models with cross-validation, including Random Forest, provided a rigorous assessment of performance but sometimes resulted in higher MAE and RMSE scores that could suggest overfitting when compared to the final Linear Regression model.

Results

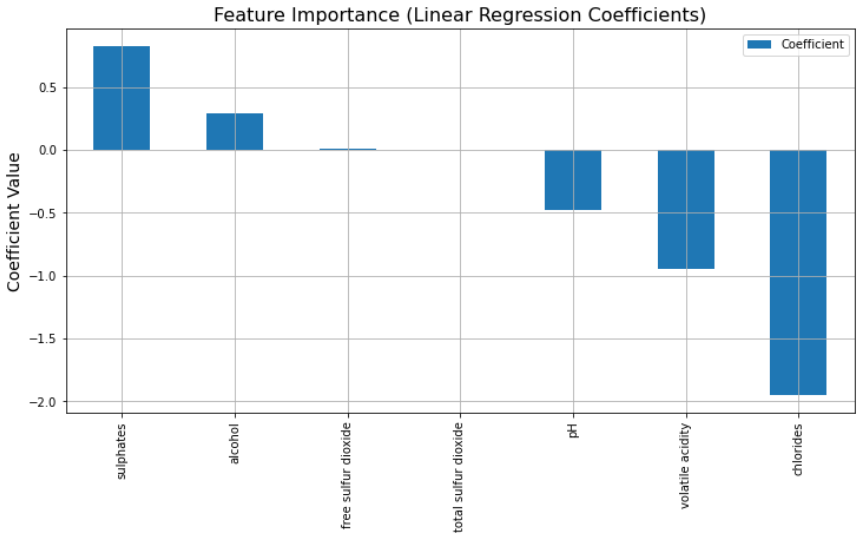

Figure 5. Feature Importance based on Correlation Coefficients

The **EDA** revealed insights into the features' distributions and identified several key features with significant correlation to the quality score. The feature importance graph from this model showed the relative significance of each feature. *Figure 5*. shows coefficients with larger magnitudes had a stronger impact on the quality score prediction, with some features like **alcohol** positively influencing quality, while others like **volatile acidity** had a negative effect.

The **Linear Regression with Backward Selection** was chosen as the final model due to its interpretability, simplicity, and the small difference in predictive accuracy between the training and test sets.

- The model achieved a Train MAE of 0.500 and a Test MAE of 0.504, with a very small difference (0.004) between them, indicating good generalization to unseen data.
- The Test MSE was 0.391, and the corresponding RMSE was 0.626, suggesting that predictions are within a reasonable range of error.

Insights

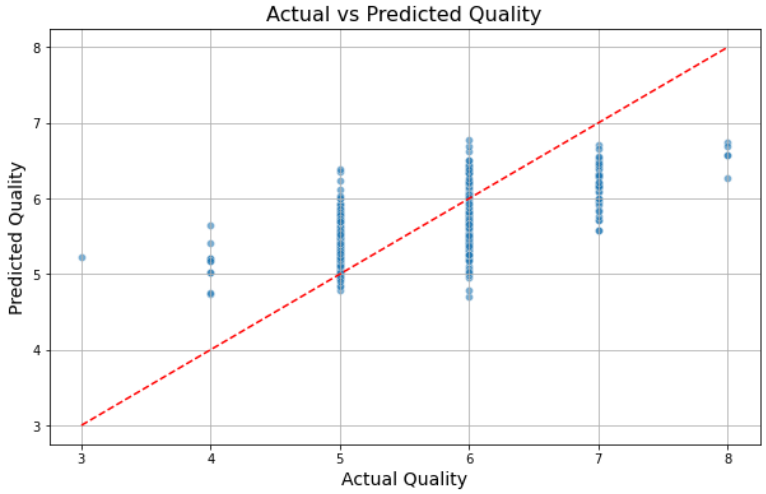

Figure 6. Scatter Plot Actual vs Predicted Quality

- This project highlights the importance of EDA, feature selection, and model evaluation in building a predictive model.
- It was observed that simpler models with fewer features can perform comparably to more complex models and are easier to interpret and deploy.
- The correlation matrix *Figure 3* indicated the presence of multicollinearity among some features, such as fixed acidity, citric acid, and density, which could affect the stability of the regression coefficients.
- The scatter plot in *Figure 6* shows that while many predictions were close to the actual values, there were variations, especially at higher quality scores, indicating potential areas for model improvement.
- The final Linear Regression model, chosen for its performance and simplicity, displayed a good understanding of the quality scores and can also handle future unseen data. However, the RMSE scores suggested room for improvement in capturing the variance in wine quality. The comparison of model results highlighted the trade-offs between model complexity and interpretability.

Conclusion and Recommendations

```python
#LR w/ backwards selection
def backward_selection(X, y, initial_list, threshold_in=0.05):
    included = list(initial_list)
    while True:
        changed = False
        model = sm.OLS(y, sm.add_constant(pd.DataFrame(X[included]))).fit()
        # use all coefficients except intercept
        pvalues = model.pvalues.iloc[1:]
        worst_pval = pvalues.max()  # worst p-value
        if worst_pval > threshold_in:
            worst_feature = pvalues.idxmax()
            included.remove(worst_feature)
            changed = True
        if not changed:
            break
    return included

selected_features = backward_selection(X_train, y_train, X_train.columns)
# Update the training and testing sets with the selected features
X_train_selected = X_train[selected_features]
X_test_selected = X_test[selected_features]
# Initialize and train the Linear Regression model
lr_model_selected = sm.OLS(y_train, sm.add_constant(X_train_selected)).fit()
# Make predictions
y_train_pred_selected = lr_model_selected.predict(sm.add_constant(X_train_selected))
y_test_pred_selected = lr_model_selected.predict(sm.add_constant(X_test_selected))
# Evaluate the model
mae_train_selected = mean_absolute_error(y_train, y_train_pred_selected)
mae_test_selected = mean_absolute_error(y_test, y_test_pred_selected)
mae_diff_selected = abs(mae_train_selected - mae_test_selected)
mse_test_selected = mean_squared_error(y_test, y_test_pred_selected)
rmse_test_selected = sqrt(mse_test_selected)
# Show the plot
plt.show()
# Print the metrics
print(f"Selected Features Linear Regression - Train MAE: {mae_train_selected:.3f}, Test MAE: {m
print(f"Test MSE: {mse_test_selected:.3f}, Test RMSE: {rmse_test_selected:.3f}")
```

Figure 7. Code of Selected Linear Regression Model

The Linear Regression model with backward selection is recommended for its advantages in interpretability and generalization. It serves as a robust starting point for predictive analysis in wine quality assessment. Correlation analysis revealed that sulphates and alcohol levels are positively correlated with wine quality, suggesting their significant roles in enhancing perceived quality. In contrast, higher levels of chlorides and volatile acidity correlate negatively, often signaling reduced wine quality.

Future work may include exploring more complex models and additional feature engineering to capture more nuanced relationships in the data. Investigate whether additional data, possibly related to subjective tasting notes or chemical compounds not included in the current dataset, could improve the predictive power of the model. There is an opportunity to further explore feature engineering, interaction terms, or non-linear modeling techniques to capture more complex relationships that may improve the model's predictive power.

References

Cortez,Paulo, Cerdeira,A., Almeida,F., Matos,T., and Reis,J.. (2009). Wine Quality. UCI Machine Learning Repository. https://doi.org/10.24432/C56S3T

GitHub Code Repository. https://github.com/aadharagarwal-hub/Wine-Quality-Analysis