STSCI 4740 Final Project

I. Introduction

This paper strives to use machine learning methods and statistical models to predict the wine quality of the red variant of Portuguese Vinho Verde wine (from a scale of 0 to 10). The dataset includes 11 features extracted from physicochemical tests: fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur oxide, total sulfur dioxide, density, pH, sulphates, and alcohol. There were 1599 observations overall.

First, to tackle this problem, feature selection was used to determine which 6 inputs were most relevant to the sensory output (the 0-10 rating). This provides the dual benefit of avoiding the multicollinearity of independent variables and enhancing the interpretability of the final model. The forward, backward, and best subset methods were used and the outputs were compared. The classification methods fit the data, specifically, the 6 variables from feature selection, were KNN, Naive Bayes, and Classification Trees. The error rates were compared and Naive Bayes was deemed to yield the best results for this dataset. The second category of machine learning methods used for this dataset was a regression, specifically a linear model, a polynomial model, and a polynomial model with interaction terms. The polynomial model resulted in the lowest error rate. Lastly, combining techniques from classification and regression, SVM regression was used to assign a class to each sample with rounding and also performed well on this dataset.

Various validation techniques were used to assess the accuracy of these different machine learning methods and analysis was conducted to evaluate the disadvantages and advantages of each. The overarching limitation for this dataset was the lack of evenly distributed samples among all of the classes, which made it difficult to train models that could make accurate predictions for certain classes.

II. Data Description

This dataset contains a total of 1599 red wine samples and 12 variables (Figure 1). The input variables are physicochemical properties. The input variables are “fixed acidity”, “volatile acidity”, “citric acid”, “residual sugar”, “chlorides”, “free sulfur dioxide”, “total sulfur dioxide”, “density”, “pH”, “sulphates”, and “alcohol,” The output variable “quality” is a score between 3 and 8 which is based on a sensory data. Due to privacy issues, the dataset does not contain wine brands or grape types. Thus, the dataset contains no qualitative variables.

fixed.acidity	volatile.acidity	citric.acid	residual.sugar	chlorides	free.sulfur.dioxide
Min. : 4.60	Min. :0.1200	Min. :0.000	Min. : 0.900	Min. :0.01200	Min. : 1.00
1st Qu.: 7.10	1st Qu.:0.3900	1st Qu.:0.090	1st Qu.: 1.900	1st Qu.:0.07000	1st Qu.: 7.00
Median : 7.90	Median :0.5200	Median :0.260	Median : 2.200	Median :0.07900	Median :14.00
Mean : 8.32	Mean :0.5278	Mean :0.271	Mean : 2.539	Mean :0.08747	Mean :15.87
3rd Qu.: 9.20	3rd Qu.:0.6400	3rd Qu.:0.420	3rd Qu.: 2.600	3rd Qu.:0.09000	3rd Qu.:21.00
Max. :15.90	Max. :1.5800	Max. :1.000	Max. :15.500	Max. :0.61100	Max. :72.00
total.sulfur.dioxide	density	pH	sulphates	alcohol	quality
Min. : 6.00	Min. :0.9901	Min. :2.740	Min. :0.3300	Min. : 8.40	Min. :3.000
1st Qu.: 22.00	1st Qu.:0.9956	1st Qu.:3.210	1st Qu.:0.5500	1st Qu.: 9.50	1st Qu.:5.000
Median : 38.00	Median :0.9968	Median :3.310	Median :0.6200	Median :10.20	Median :6.000
Mean : 46.47	Mean :0.9967	Mean :3.311	Mean :0.6581	Mean :10.42	Mean :5.636
3rd Qu.: 62.00	3rd Qu.:0.9978	3rd Qu.:3.400	3rd Qu.:0.7300	3rd Qu.:11.10	3rd Qu.:6.000
Max. :289.00	Max. :1.0037	Max. :4.010	Max. :2.0000	Max. :14.90	Max. :8.000

Figure 1: Summary of the dataset

This dataset can be viewed as a classification or regression task. For classification, 12 class labels were assigned for each quality score and aim to predict the target class for the test

data. For regression, the quality score was considered a continuous variable and aim to predict a score between 0 and 12 for the test data.

	fixed.acidity	volatile.acidity	citric.acid	residual.sugar	chlorides	free.sulfur.dioxide
1	7.4	0.70	0.00	1.9	0.076	11
2	7.8	0.88	0.00	2.6	0.098	25
3	7.8	0.76	0.04	2.3	0.092	15
4	11.2	0.28	0.56	1.9	0.075	17
5	7.4	0.70	0.00	1.9	0.076	11
6	7.4	0.66	0.00	1.8	0.075	13

	total.sulfur.dioxide	density	pH	sulphates	alcohol	quality
1	34	0.9978	3.51	0.56	9.4	5
2	67	0.9968	3.20	0.68	9.8	5
3	54	0.9970	3.26	0.65	9.8	5
4	60	0.9980	3.16	0.58	9.8	6
5	34	0.9978	3.51	0.56	9.4	5
6	40	0.9978	3.51	0.56	9.4	5

Figure 2: Head of the dataset

III. Variable selection

When applying models, variable selection is useful because it discards irrelevant predictors and leads to simpler models that are easier to interpret and that usually perform better. On the other hand, complex models may overfit the data and produce a high variance. To determine the number of variables in a subset, plots of objective criteria like R-Square, C_P, adjusted R-Square, and AIC were compared for different numbers of variables. The plots show a small change in values when the number of variables exceeds 6. In order to achieve an interpretable yet computationally efficient subset, having 6 variables in a subset would produce the smallest C_P and AIC and the largest R-Square and adjusted R-Square values. Also, having more than six variables in a subset may overfit the data.

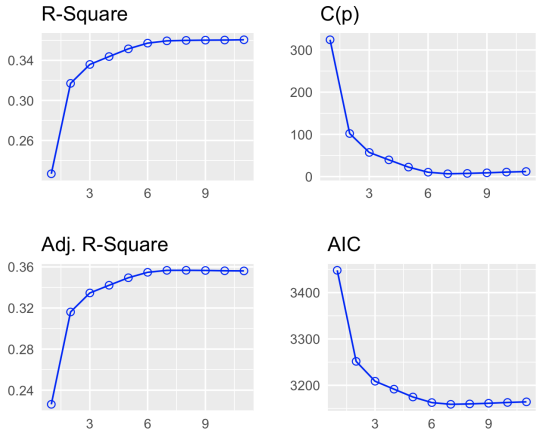

Figure 3: Plots of the four criteria for selection

The best subset, forward, and backward methods were applied to find the best six-variables model. The backward method was feasible because the number of datasets is greater than the number of predictors. Three subset selection methods (best, forward, and backward) indicate that the best six-variable model is (quality ~ volatile.acidity + chlorides + total.sulfur.dioxide + pH + sulphates + alcohol). If the methods produced different models, the model from the best subset selection would've been chosen because it guarantees finding the best

possible model. This wasn't necessary because all methods produced the same model, so the aforementioned best six-variable model was used to eliminate unnecessary predictors.

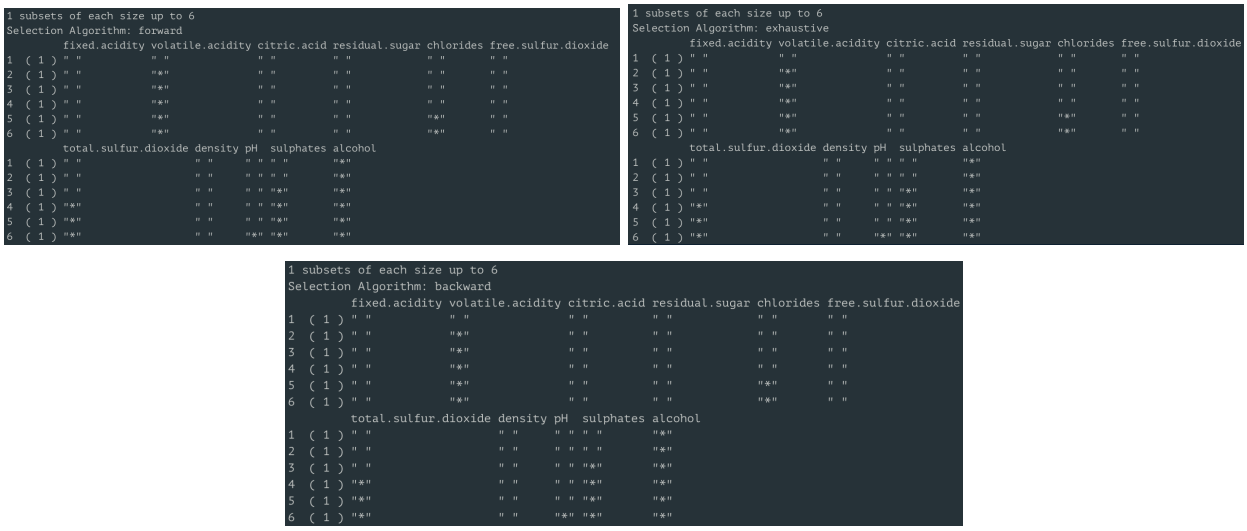

Figure 3: Forward selection (top left), best subset selection (top right), backward selection (bottom)

IV. Method 1: Classification

quality <int>	n <int>
3	10
4	53
5	681
6	638
7	199
8	18

Figure 4: number of samples per class

First, the data was filtered to only contain the variables deemed important via feature selection, so only 6 predictors remained. Next, samples were split into a training and test set, with 80% and 20% of the data going into each subset respectively. The quality column of the dataframe only ranges from 3 to 8, so the classification model doesn't have the training data necessary to accurately predict the rating of wines categorized as a 0-2 or a 9-10. Furthermore, due to the imbalanced nature of the dataset, there are significantly fewer training examples available for wines of classes 3, 4, and 8. Most of the predictions were made for classes 5, 6, and 7 as seen in the confusion matrices and analysis section. Although the data was randomly split, there was representation from each of the classes between 3-8 in both the training and test sets.

The first method, KNN, is non-parametric and can handle multi-class classification problems. Since there are multiple classes and the decision boundary is likely non-linear, in theory, this could be a suitable model if the assumption holds that samples with similar labels exist in close proximity. Using 5-fold cross-validation, K=9 was determined to be the number of neighbors that yielded the lowest RMSE. KNN with k-9 was then fitted on all of the training data and used to make predictions on the test set, with an error rate of .46875.

KNN is very sensitive to scaling, so the above procedure was repeated on a dataset with normalized predictors, so they all have a mean of 0 and a standard deviation of 1. If the

predictors are un-normalized for KNN, variables on a large scale like total.sulfur.dioxide will affect the distance between data points significantly more than a variable like volatile.acidity would. As expected, the test error dropped from .46875 to .4, which means the accuracy rate is around 60%.

Naive Bayes makes the assumption all features are independent. While it's unknown if this holds in the training dataset, feature selection likely eliminated any multicollinearity in the predictors. Again, this is a non-linear classification algorithm suited to handling multiple classes. Fitting a Naive Bayes model on the training data, using it to make predictions on the test set, and comparing it with the actual test set data resulted in an error rate of .3875, which means the accuracy rate is slightly higher at 61.25%.

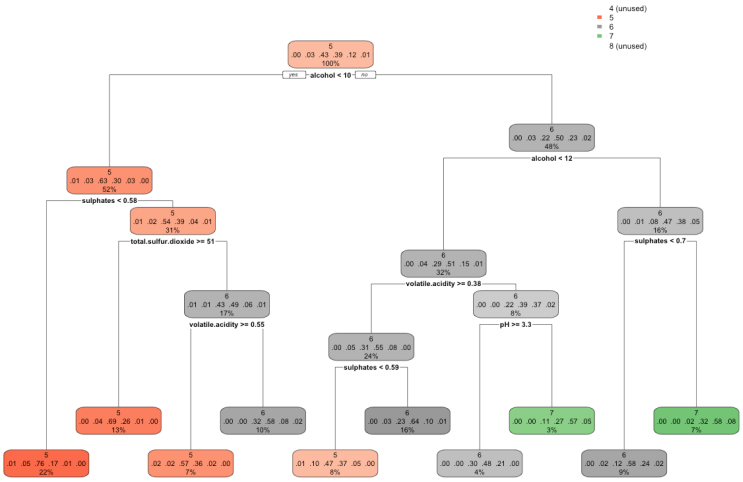

Figure 5: Classification Tree

Classification trees are very interpretable and easy to visualize due to their structure. After fitting on the training data and making predictions, the error rate is the highest compared to the previous two models at .415625 and the accuracy rate is 58.4375%. The classification tree didn't even include 3, 4, or 8 as one of the leaves, therefore incorrectly predicting any test samples from those 3 classes. This result is likely due to the model trying to achieve the lowest test error, thus always predicting a 5, 6, or 7 due to the imbalanced nature of the dataset.

V. Method 2: Regression

The first regression model that was built is a linear model with the six variables selected by best subset selection (quality ~ volatile.acidity + chlorides + total.sulfur.dioxide + pH + sulfates + alcohol). The linear regression has a leave-one-out cross-validation error (LOOCV) of approximately 42.36%.

```
Call:
glm(formula = quality ~ volatile.acidity + chlorides +
total.sulfur.dioxide +
    pH + sulphates + alcohol, family = gaussian, data =
winequality.red)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.60575  -0.35883  -0.04806   0.46079   1.95643

Coefficients:
                       Estimate Std. Error t value Pr(>|t|)
(Intercept)           4.2957316  0.3995603  10.751  < 2e-16
volatile.acidity     -1.0381945  0.1004270 -10.338  < 2e-16
chlorides            -2.0022839  0.3980757  -5.030 5.46e-07
total.sulfur.dioxide -0.0023721  0.0005064  -4.684 3.05e-06
pH                   -0.4351830  0.1160368  -3.750 0.000183
sulphates             0.8886802  0.1100419   8.076 1.31e-15
alcohol               0.2906738  0.0168108  17.291  < 2e-16
```

Figure 6: Linear Regression Model Summary

Since some of the predictors seem to have a non-linear relationship with quality, adding polynomial terms to the regression will reduce the error rate.

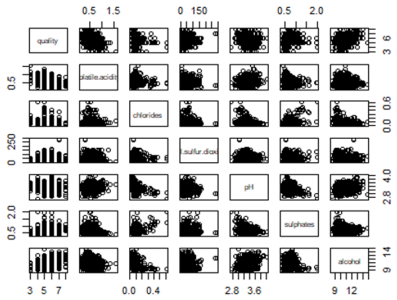

Figure 7: Scatterplots of each pair of variables

The polynomial regression model was found through a process that is similar to the forward selection. First, a linear model was created with all variables. Then, models were created with all variables, but with one variable changing at a time. Models were made by changing one variable to either a 1st, 2nd, 3rd, or 4th-degree polynomial or a logarithmic term, and the LOOCV error of each model was found. Once the degree with the smallest error was identified, the model was changed to include that variable with its best polynomial degree. Then, the process was repeated for the next variable keeping the changes to the model that were made in the previous step. This continued until all variables were tested to find the polynomial degree that produced the smallest error given the other variables' polynomial terms that were already in the model.

The polynomial constructed in this way that had the lowest LOOCV error is when sulfates, volatile.acidity, and total.sulfur.dioxide are third-degree polynomial terms, pH is a second-degree polynomial, alcohol is a fourth-degree polynomial, and chlorides is a first-degree polynomial. This model had a LOOCV error of 40.42% which is slightly lower than the linear regression.

This way of designing a polynomial regression is a greedy algorithm because it is not exploring all possible models. It is possible that there are better polynomial models, however, it

would be very computationally expensive to check all models. Since 5 different transformations were tested (1st, 2nd, 3rd, and 4th degree and logarithmic) and there were 6 predictors, there would be 5^6 = 15625 possible models to check.

```
summary(poly.regression)

##
## Call:
## glm(formula = quality ~ poly(volatile.acidity, 3, raw = T) +
##     chlorides + poly(total.sulfur.dioxide, 3, raw = T) + poly(pH,
##     2, raw = T) + poly(sulphates, 3, raw = T) + poly(alcohol,
##     4, raw = T), family = gaussian, data = winequality.red)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.73618  -0.39843  -0.02432   0.43549   1.94906
##
## Coefficients:
##                                          Estimate Std. Error t value Pr(>
|t|)
## (Intercept)                             -1.509e+02  7.101e+01  -2.125 0.03
3780
## poly(volatile.acidity, 3, raw = T)1     -2.157e+00  1.013e+00  -2.130 0.03
3358
## poly(volatile.acidity, 3, raw = T)2      2.375e+00  1.547e+00   1.534 0.12
5113
## poly(volatile.acidity, 3, raw = T)3     -1.256e+00  7.275e-01  -1.726 0.08
4560
## chlorides                               -1.530e+00  4.011e-01  -3.815 0.00
0141
## poly(total.sulfur.dioxide, 3, raw = T)1  5.036e-03  2.559e-03   1.968 0.04
9233
## poly(total.sulfur.dioxide, 3, raw = T)2 -9.288e-05  2.877e-05  -3.228 0.00
1272
## poly(total.sulfur.dioxide, 3, raw = T)3  2.943e-07  8.247e-08   3.568 0.00
0370
## poly(pH, 2, raw = T)1                    5.323e+00  2.867e+00   1.856 0.06
3583
## poly(pH, 2, raw = T)2                   -8.899e-01  4.304e-01  -2.068 0.03
8841
## poly(sulphates, 3, raw = T)1             8.367e+00  1.454e+00   5.754 1.04
e-08
## poly(sulphates, 3, raw = T)2            -6.815e+00  1.528e+00  -4.460 8.79
e-06
## poly(sulphates, 3, raw = T)3             1.714e+00  4.764e-01   3.598 0.00
0331
## poly(alcohol, 4, raw = T)1               5.504e+01  2.537e+01   2.169 0.03
0194
## poly(alcohol, 4, raw = T)2              -7.762e+00  3.384e+00  -2.293 0.02
1956
## poly(alcohol, 4, raw = T)3               4.843e-01  1.994e-01   2.429 0.01
5269
## poly(alcohol, 4, raw = T)4              -1.122e-02  4.379e-03  -2.563 0.01
0481
```

Figure 8: Polynomial Regression Model Summary

Next, interaction terms were added to the model to see if it reduces the error. First, all 15 possible interaction terms were added and their p-values were found to determine which terms were significant in predicting quality. The only interaction terms that were significant at alpha = 0.05 were volatile.acidity and total.sulfur.dioxide, total.sulfur.dioxide and sulphates, chlorides and alcohol, and sulphates and alcohol. These four interaction terms were added to the previous model and the LOOCV decreased to 38.83%. Similar to the way the polynomial was constructed, not all interaction models were checked because it would be very computationally expensive. A model can have any combination of interaction terms and any number of terms from 0-15.

VI. SVM Regression

SVM regression was used to classify the data in accordance with the methods used in the relevant paper referenced on the website containing the dataset. For this SVM model, a Gaussian RBF kernel and a gamma value of 2^{-19} was used and predictions were rounded to the nearest class. Like KNN, the normalized version of the dataset was used. This algorithm is well suited for non-linear data and as expected, yields the lowest error rate out of all the classification methods of 36.25%. Instead of rounding, if the tolerance is further increased to accept either of the two closest classes (an output of 5.1 would be counted as correct if the actual label is either 5 or 6) then the error rate decreases to just 10%. .

VII. Comparison & Analysis

The LOOCV was chosen as the validation method for regression because it is an estimate of the test MSE. Although this method is computationally expensive, it allows the training set and testing sets to both be almost the same as the entire data set. It also has no randomness, so it is the same every time the program is run.

One problem with using a regression on this data is that quality is a discrete variable while regression produces numbers in between the values on the quality scale. The LOOCV is an estimate of the test MSE, which is the average squared difference between the quality predicted by the model and the actual data. Since regression usually produces a number with a decimal, there will be a difference even if the numbers are very close. This could explain why classification may have a smaller error than the regression models. One solution to this could be splitting the data into a training and testing set, choosing the polynomial regression and interaction terms just based on the training set. Then, when making the predictions with the test set, the predicted quality could be rounded to the nearest whole number before being compared to the actual quality value. However, the process can't be done on the polynomial model that was constructed because it was already made with all the data. Additionally, splitting the data isn't stable, meaning the model that is created could be different with each different way of splitting the data.

Confusion matrices can be used to more extensively analyze the performance of a classifier. Using the one for Naive Bayes as an example, comparing the predictions with the actual labels from the test set, it's evident that all samples from class 3 and class 8 were classified incorrectly. This is likely due to the imbalanced dataset causing there to be a lack of training data available for these classes. Within the other classes (4, 5, 6, and 7), the error rates were 82%, 29.6%, 37%, 50%. For reference, if using random guessing, the error rate overall would be about 80%. Classes 5 and 6 have around 13 times more data than class 4 and three times as many training points as class 7, and subsequently are able to be predicted with significantly higher accuracy. There weren't any classification methods that significantly outperformed the others, with all of the accuracy rates hovering at around 60%. Depending on how much tolerance is allowed when it comes to classification, however, SVM regression may hold the most potential for classifying this dataset.

	labels					
nb.class	3	4	5	6	7	8
3	0	0	0	0	0	0
4	2	2	2	3	0	0
5	2	5	88	37	1	0
6	0	2	34	85	19	1
7	0	1	0	8	21	2
8	0	1	1	2	1	0

Figure 9: Naive Bayes confusion matrix

VIII. Conclusion & Limitations

In our paper, both classification and regression models were used to predict red wine quality because the output variable is a score between 3 and 8. Variable selection using forward, backward, and best subset methods was used to minimize overfitting and increase efficiency. A linear model with the six variables (quality ~ volatile.acidity + chlorides + total.sulfur.dioxide + pH + sulfates + alcohol) was determined to perform the best.

Then, classification methods were used to fit the data. The three methods we used are KNN, Naive Bayes, and Classification Tree. Among the three methods, Naive Bayes produced the smallest test error rate of 0.3875. Three regression methods were also used to fit the data: a linear model, a polynomial model, and a polynomial model with interaction terms. Among the three models, the polynomial model with four interaction terms produced the smallest test error rate of 0.3883. Lastly, an SVM regression model was used to classify the samples and had the smallest test error of all at .3625.

Since all of the test error rates are very similar, it's difficult to determine which model is the best. As previously mentioned, this dataset is naturally more suited to classification methods because the response isn't a continuous variable. Either Naive Bayes or SVM regression (rounded for classification) may the best methods to make predictions on this dataset.

One limitation of the study is the relationship between the input variables and and the output variable, quality. While the input variables are physicochemical properties, the output variable is sensory data that relies mainly on human experts. The relationship between physicochemical and sensory analysis is complex and subjective. This made it difficult to identify trends and patterns, which is reflected in the relatively high test error rate. Because complex models may overfit the data, the number of parameters used in the model was reduced to get good predictive accuracy. Given that there are 5 classes and a base error rate of 20% resulting from random guessing, however, it does appear that machine learning methods do contribute insights to this red wine dataset.

IX. R Code

[1] Variable Selection:
https://drive.google.com/file/d/1dJES1mmu4Mtcjyy_oGVishHxTb-amEgB/view?usp=sharing

[2] Classification:
https://drive.google.com/file/d/1NRDV2ULYELCLUAaYZUQhIs7B6jAymLEN/view?usp=share_link

[3] Regression:
https://drive.google.com/file/d/1bEDosSOAMIBzok5KS78LpK3SeH6dEpq_/view?usp=sharing

X. Citation

Cortez, Paulo, et al. "Modeling wine preferences by data mining from physicochemical properties." *Decision Support Systems*, vol. 47, no. 4, 2009, pp. 547-553, https://doi.org/10.1016/j.dss.2009.05.016. Accessed 4 Dec. 2022.

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