BAN 620-01 Data Mining

Spring Semester 2023

Group Project

Supervised Learning Methods to Predict the Quality of Red Wine

May 14, 2023

Apurva Vaze (vaaf8z)

Neenu Jose (wg8684)

Pallavi Hunswadkar (ls7546)

Ruchika Vasudev Balani (ac9968)

Swetha Srinivasan (cx2194)

Table of Contents

Summary

This project aimed to predict the quality of red wine using machine learning algorithms. The project used a red wine quality dataset from Kaggle and preprocessed the data by removing duplicates and converting the numeric quality column to categorical labels (low, medium, and high).

Three machine learning algorithms were used to predict the quality of wine – classification tree, ordinal logistic regression, and neural networks. The evaluation metric used to compare the performance of the algorithms was confusion matrix. After analyzing the results, the main finding was that the classification tree algorithm performed better than the other algorithms in terms of accuracy.

Overall, the project demonstrates the potential of machine learning algorithms in predicting red wine quality and highlights the effectiveness of the classification tree algorithm for this particular task. The findings suggest that further research could be done to refine the classification tree algorithm and improve the accuracy of the predicting low- and high-quality wines even further.

Introduction

Wine has been a cherished beverage for centuries, enjoyed by people across the globe for its nuanced flavors and aromas. However, despite the popularity of wine, its quality evaluation remains a daunting task for both experts and novices alike. Traditionally, wine quality has been assessed through sensory evaluation by trained panels or individual tasters, a process that is time-consuming, expensive, and subjective.

In recent years, there has been a growing interest in using data-driven approaches, such as machine learning, to predict wine quality based on measurable physicochemical attributes. Such approaches

have the potential to automate wine quality assessment, provide objective and consistent results, and enable large-scale analysis of wine quality trends.

In this report, we will explore a dataset of red wine samples which contains several physicochemical attributes of the wines, as well as their quality rating on a scale from 3 to 8. To achieve this goal, we will first explore the dataset and perform data cleaning and preprocessing. Then, we will split the data into training and testing sets and train several supervised learning algorithms, such as decision trees, logistic regression, and neural networks. We will evaluate the performance of each model on the testing data using metrics such as confusion matrices and select the best-performing one.

Overall, this report aims to showcase the potential of supervised learning and neural networks for wine quality prediction and provide a unique perspective on the physicochemical factors that contribute to red wine quality.

Main Chapter

Step 1 – Understanding the Problem Statement

The wine dataset contains various chemical details about the wine, such as acidity level, alcohol content, sweetness, and a rating of its quality. Our aim is to predict the quality of the wine based on these chemical properties.

The intended audience for this project includes winemakers, wine experts, and wine sellers. They may use the analysis results to determine which chemical features are important in determining the quality of red wine.

The predictions of this project can help identify the right price for the right quality of wine. In addition, the results can also help stakeholders devise appropriate marketing plans according to the quality of wines.

Using prediction models, we can also develop rules to estimate the quality of wine and ultimately improve the wine-making process. Whether this analysis is a one-time event, or a continuous effort depends on the goals of the stakeholders. If the objective is to make better wine, ongoing analysis may be necessary.

Step 2 - Obtain Data for Analysis

This dataset was found from Kaggle with 1599 records and 12 variables namely fixed acidity, volatile acidity, citric acid, residual sugar, chloride, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol, and quality (*Red Wine Quality*, n.d.).

```
Number of rows and columns in data set: (1599, 12)
```

Description of Dataset

Variables	**Description**
Fixed Acidity	Volume of acids in a sample measured in grams per cubic decimeter (g/dm^3)
Volatile Acidity	Volume of acetic acid in a sample measured by (g/dm^3)
Citric Acid	Used by winemakers in acidification to boost the wine's total acidity measured by (g/dm^3)
Residual sugar	Amount of sugar left after fermentation measured by (g/dm^3)
Chlorides	Amount of salt in wine measured by (g/dm^3)
Free Sulfur Dioxide	Amount of unbound sulfur dioxide that inhibits microbial growth and oxidation measured by milligram per cubic decimeter (mg/dm^3)

Total Sulfur Dioxide	Total amount of sulfur dioxide in wine – both free and bound - measured by (mg/dm^3)
Density	Density of the wine measured by gram per cubic centimeter (g/cm^3)
pH	pH describes how acidic or basic a wine is
Sulphates	Contributes to sulphur dioxide gas, measured by (g/dm^3)
Alcohol	Percent of alcohol in wine
Quality	Categorical value on a scale between 3 and 8, higher values denote higher quality

(Cortez et al., 2009)

The data was imported as a pandas dataframe named winequality_df. The first five and last five records from this dataset are presented below.

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	5
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	6
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
1594	6.2	0.600	0.08	2.0	0.090	32.0	44.0	0.99490	3.45	0.58	10.5	5
1595	5.9	0.550	0.10	2.2	0.062	39.0	51.0	0.99512	3.52	0.76	11.2	6
1596	6.3	0.510	0.13	2.3	0.076	29.0	40.0	0.99574	3.42	0.75	11.0	6
1597	5.9	0.645	0.12	2.0	0.075	32.0	44.0	0.99547	3.57	0.71	10.2	5
1598	6.0	0.310	0.47	3.6	0.067	18.0	42.0	0.99549	3.39	0.66	11.0	6

Step 3 – Explore, Clean and Preprocess Data

Column titles were modified to remove any trailing spaces or double worded names.

```
Modified column titles with no space and one word for titles:

Index(['fixed_acidity', 'volatile_acidity', 'citric_acid', 'residual_sugar',
       'chlorides', 'free_sulfur_dioxide', 'total_sulfur_dioxide', 'density',
       'pH', 'sulphates', 'alcohol', 'quality'],
      dtype='object')
```

The dataframe was found to have 240 duplicate records (pictured below), which were removed from the dataset. Following this, the winequality_df dataframe had 1359 observations and 12 variables.

	fixed_acidity	volatile_acidity	citric_acid	residual_sugar	chlorides	free_sulfur_dioxide	total_sulfur_dioxide	density	pH	sulphates	alcohol	quality
4	7.4	0.700	0.00	1.90	0.076	11.0	34.0	0.99780	3.51	0.56	9.4	5
11	7.5	0.500	0.36	6.10	0.071	17.0	102.0	0.99780	3.35	0.80	10.5	5
27	7.9	0.430	0.21	1.60	0.106	10.0	37.0	0.99660	3.17	0.91	9.5	5
40	7.3	0.450	0.36	5.90	0.074	12.0	87.0	0.99780	3.33	0.83	10.5	5
65	7.2	0.725	0.05	4.65	0.086	4.0	11.0	0.99620	3.41	0.39	10.9	5
...	...	...	...	...	...	...	...	...	...	...	...	...
1563	7.2	0.695	0.13	2.00	0.076	12.0	20.0	0.99546	3.29	0.54	10.1	5
1564	7.2	0.695	0.13	2.00	0.076	12.0	20.0	0.99546	3.29	0.54	10.1	5
1567	7.2	0.695	0.13	2.00	0.076	12.0	20.0	0.99546	3.29	0.54	10.1	5
1581	6.2	0.560	0.09	1.70	0.053	24.0	32.0	0.99402	3.54	0.60	11.3	5
1596	6.3	0.510	0.13	2.30	0.076	29.0	40.0	0.99574	3.42	0.75	11.0	6

240 rows × 12 columns

```
Shape of the dataframe after removing duplicate records is
 (1359, 12)
```

The dataframe did not have any missing values.

```
Number of missing records in winequality_df
fixed_acidity           0
volatile_acidity        0
citric_acid             0
residual_sugar          0
chlorides               0
free_sulfur_dioxide     0
total_sulfur_dioxide    0
density                 0
pH                      0
sulphates               0
alcohol                 0
quality                 0
dtype: int64
```

The float64 datatypes represents continuous numeric variables. Variable datatypes were found to be numeric for all variables including quality. However, for this project quality of the wine is being converted to a categorical variable. Therefore, the ordered categorical variable is represented as objects that have three classes – low ranging from 3-4, medium from 5-6 and high from 7-8.

```
Variable data types before
fixed_acidity           float64
volatile_acidity        float64
citric_acid             float64
residual_sugar          float64
chlorides               float64
free_sulfur_dioxide     float64
total_sulfur_dioxide    float64
density                 float64
pH                      float64
sulphates               float64
alcohol                 float64
quality                   int64
dtype: object
```

```
Variable data types after
fixed_acidity           float64
volatile_acidity        float64
citric_acid             float64
residual_sugar          float64
chlorides               float64
free_sulfur_dioxide     float64
total_sulfur_dioxide    float64
density                 float64
pH                      float64
sulphates               float64
alcohol                 float64
quality_category       category
dtype: object
```

The number of records in each category after conversion is as follows.

```
Number of records in each category:
medium    1112
high       184
low         63
```

Step 5 – Determine Data Mining Task

Our objective is to classify the quality of red wine which is an ordinal variable represented by quality_category using 11 predictor variables. Since there is a specific target to predict, this task will be considered supervised learning. In order to predict the quality of red wine, the project explores the use of three data mining models – classification tree, logistic regression and neural networks. Specifically, since we have three categories (low, medium, and high), we will use ordinal logistic regression to determine the quality of red wine.

To utilize logistic regression for multiclass classification, the categories must be converted into numeric values such as dummy variables (0, 1, 2, etc.), as the logistic function's formula represents the probability of a multiple outcome variable as numeric values, not alphanumeric categories like 'low', 'medium', and 'high'. This project will therefore convert quality variable at two steps – once to convert from numerical to categorical in three levels (high, medium and low) and once to convert alphabetical labels to numeric dummy variables. The three-level dummy variables will be called *quality_category_int* in the dataframe, and will be used for both logistic regression and neural network model. The results will be compared to identify the most effective model.

```
Predictors (X):
['fixed_acidity', 'volatile_acidity', 'citric_acid', 'residual_sugar', 'chlorides', 'free_sulfur_dioxide', 'total_sulfur_dioxid
e', 'density', 'pH', 'sulphates', 'alcohol']
Outcome variable (y):
quality_category
```

```
Predictors (X):
['fixed_acidity', 'volatile_acidity', 'citric_acid', 'residual_sugar', 'chlorides', 'free_sulfur_dioxide', 'total_sulfur_dioxid
e', 'density', 'pH', 'sulphates', 'alcohol']
Outcome variable (y):
quality_category_int
```

Step 6 – Partitioning Data

We will partition the dataset using the train_test_split() function from the scikit-learn library. The dataset was partitioned 60-40 into train and test segments. With 60% of the dataset, the training partition had 815 records and 11 predictors including fixed_acidity, volatile_acidity, citric_acid, residula_sugar, chlorides, free_sulphur_dioxide, total_sulphur_dioxide, density, pH, sulphates, and alcohol. The remaining 40% – 544 records and 11 predictors – was in the validation partition.

For the classification tree model, the outcome variable was quality_category with three levels high, medium, and low. For logistic regression and neural networks models the outcome variable was quality_category_int. To accommodate this difference the train_test_split function was used twice,

once for classification tree (train_X_ct) and once for both logistic and neural nets models (train_X).

```
Training set dimensions:
(815, 11) (815,)
Validation set dimensions:
(544, 11) (544,)
```

Step 7 – Data Mining Techniques

The classification tree was chosen as one of the methods since it is visually represented and is easy to understand. Using classification and regression trees (CART) results in a set of rules that could be used for future predictions, which for executives would be a quick way to estimate the quality of wine given some characteristics.

Since the output of the ordinal logistic regression will be in the form of an odds ratio, it will tell us how the odds of a wine being in a higher quality category can change with a unit increase in one of the independent variables, holding all other independent variables constant.

In the case of the red wine quality dataset, a neural network can be appropriate because it has the ability to model complex relationships between the independent variables and the dependent variable quality.

Step 8 – Algorithm and Measures

Classification Tree

The GridSearchCV() algorithm was initiated with random numbers on the training partition to find the optimal parameters for a decision tree. Initial parameters were identified for a classification tree with the highest possible accuracy rating of 82.82% and the lowest possible homogeneity. The

tree was found to have maximum depth of 10 layers, with minimum decrease in impurity at 0.01 and 20 minimum records in a node to consider splitting.

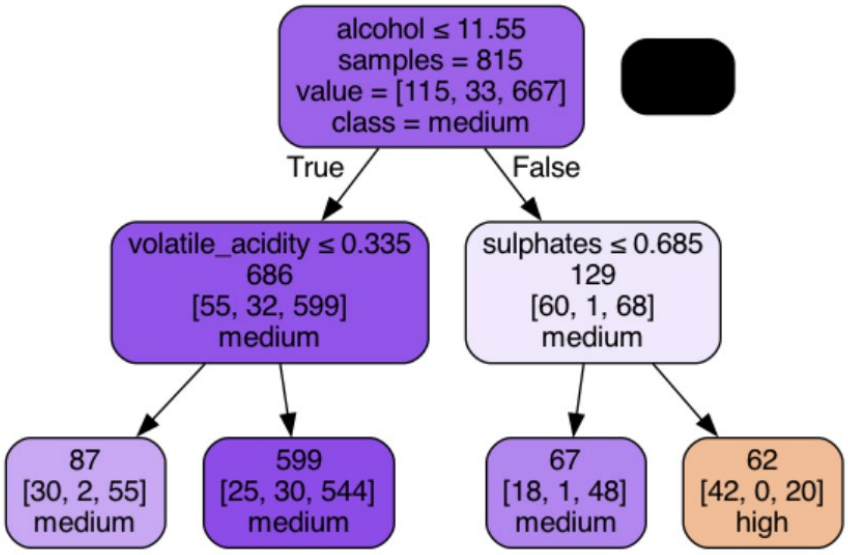
```
Initial score:0.8282
Initial parameters:  {'max_depth': 10, 'min_impurity_decrease': 0.01, 'min_samples_split': 20}
```

Using these initial parameters, GridSearchCV() was run once again, which resulted in the ideal classification tree with highest possible accuracy of 83.31% and lowest possible homogeneity. This tree had maximum depth of 2 layers, 0 minimum decrease in impurity and a minimum of 10 records before a node can be split further.

```
Improved score:0.8331
Improved parameters:  {'max_depth': 2, 'min_impurity_decrease': 0, 'min_samples_split': 10}
```

The classification tree was developed using the improved parameters found above using GridSearchCV(). The outcome variable in the tree had three classes high, low, medium (in alphabetical order). The tree has two levels of split and seven nodes – of which three are decision nodes and four are terminal.

```
Classes: high, low, medium
Best Classification Tree with Grid Search
```



It is observed that the numerous nodes have moderately high heterogeneity – including some of the terminal nodes. Perhaps the factor limiting growth of the tree here is the number of levels of split.

Interpreting the tree, the quality of a wine is 83% likely to be high when alcohol content is > 11.55, and sulphates > 0.685. It is interesting to note that using just two or three characteristics of wine, the decision tree is able to predict its quality with relatively high confidence.

Confusion matrices for training and validation partition are presented below. 0 represents high wine quality, 1 represents low quality and 2 medium wine quality. In training partition 126 records out of 815 (15.46%) were misclassified, and the model has 84.54% accuracy. Its performance in validation partition is similar to that in training partition, where 88 out of 544 records were misclassified (16.17%), regardless the model has a high accuracy rate of 83.82%. On comparing the accuracy rates between training and validation partition, it can be safely concluded that the classification tree model does not overfit the data.

```
Performance Measures for Training Partition using Classification Tree
Confusion Matrix (Accuracy 0.8454)

       Prediction
Actual   0   1   2
     0  42   0  73
     1   0   0  33
     2  20   0 647

Performance Measures for Validation Partition using Classification Tree
Confusion Matrix (Accuracy 0.8382)

       Prediction
Actual   0   1   2
     0  26   0  43
     1   0   0  30
     2  15   0 430
```

Ordinal Logistic Regression

Using LogisticIT() from mord library, ordinal logistic regression model was trained. The following intercepts and coefficients were obtained.

```
Ordinal Logistic Regression
  Intercepts [0.595 7.557]
  Coefficients [ 1.120e-01 -4.133e+00 -3.490e-01  3.000e-03 -4.394e+00  1.100e-02
 -4.000e-03 -5.560e-01 -1.604e+00  4.260e+00  9.230e-01]
```

The intercepts and coefficients can be represented in a mathematical equation below. The first intercept represents the log-odds of being in the "low" quality category compared to being in the "medium" or "high" quality categories, and the second intercept represents the log-odds of being in the "low" or "medium" quality categories compared to being in the "high" quality category.

$$P(\text{quality} = 0 \text{ (low)}) = P(Y=0) = \frac{1}{1 + e^{-(0.595+0.1120x1-4.133x2-0.3490x3+0.0030x4-4.394x5+0.0110x6-0.0040x7-0.5560x8-1.604x9+4.260x10+0.9230x11)}}$$

$$P(\text{quality} = 1 \text{ (medium)}) = P(Y \leq 1) - P(Y=0) = \frac{1}{1 + e^{-(7.557+0.1120x1-4.133x2-0.3490x3+0.0030x4-4.394x5+0.0110x6-0.0040x7-0.5560x8-1.604x9+4.260x10+0.9230x11)}} - \frac{1}{1 + e^{-(0.595+0.1120x1-4.133x2-0.3490x3+0.0030x4-4.394x5+0.0110x6-0.0040x7-0.5560x8-1.604x9+4.260x10+0.9230x11)}}$$

$$P(\text{Severity} = 2 \text{ (high)}) = 1 - P(Y \leq 1) = 1 - \frac{1}{1 + e^{-(7.557+0.1120x1-4.133x2-0.3490x3+0.0030x4-4.394x5+0.0110x6-0.0040x7-0.5560x8-1.604x9+4.260x10+0.9230x11)}}$$

The coefficients represent the change in the log-odds of moving up one category (i.e., from "low" to "medium", or from "medium" to "high") for each unit increase in the predictor variable.

For example, let's say the coefficient for predictor variable x1 is 0.1120. This means that for each unit increase in x1, the log-odds of moving up one category (i.e., from "low" to "medium", or from "medium" to "high") increases by 0.1120.

The following table shows the classification results for the first 10 records in the validation dataset. The "Actual" column shows the true category for each record, while the "Classification" column shows the predicted category based on the model. The "P(0)", "P(1)", and "P(2)" columns show the predicted probabilities of each record being in the "low", "medium", and "high" quality categories, respectively.

Classification for First 10 Records in Validation Data Set

	Actual	Classification	P(0)	P(1)	P(2)
473	1	1	0.0025	0.7263	0.2711
1376	1	1	0.1389	0.8553	0.0058
533	1	2	0.0003	0.2629	0.7368
200	2	1	0.0024	0.7138	0.2838
268	1	1	0.0253	0.9395	0.0352
210	1	2	0.0008	0.4483	0.5509
906	1	1	0.0106	0.9082	0.0812
1093	2	2	0.0005	0.3610	0.6385
947	2	1	0.0010	0.5072	0.4918
867	1	1	0.0035	0.7851	0.2114

For each record, the model has predicted the class that has predicted probability of more than 0.5. It is evident that the model accurately predicted 6 out of 10 records while incorrectly predicting the remaining 4 records. As an example, for record number 200, the expected value of 2 (medium) did not match the predicted value of 1 (low).

The accuracy measure for training partition is approximately 84%, and for validation partition is 82%. Since there is no significant difference in accuracy of training and validation partition it can be inferred that there is no overfitting. The misclassification rate for training partition is approximately 16% (127 out of 815 records) and for validation it is 18% (99 out of 544 records).

```
Accuracy Measure for Training Partition for Ordinal Logistic Model
Confusion Matrix (Accuracy 0.8442)

       Prediction
Actual   0   1   2
     0   1  32   0
     1   1 645  21
     2   0  73  42

Accuracy Measure for Validation Partition for Ordinal Logistic Model
Confusion Matrix (Accuracy 0.8180)

       Prediction
Actual   0   1   2
     0   1  28   1
     1   0 423  22
     2   0  48  21
```

Neural Network Model

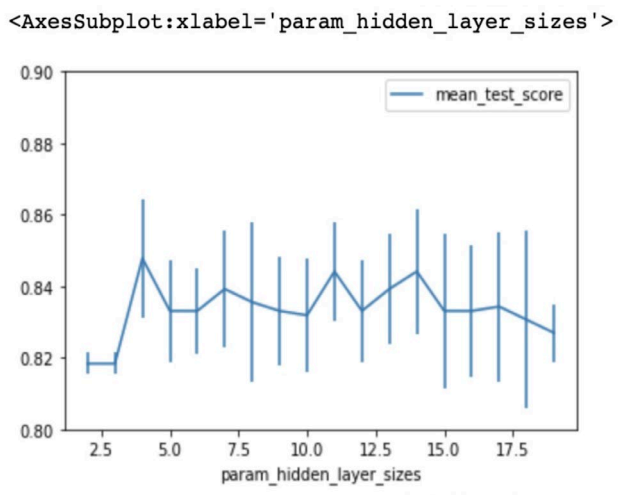
GridSearchCV() function is used to identify the best number of nodes for the hidden layer as 4 with improved score of 0.8479. The excessive number of iterations or hidden layers used in developing a neural network model can often lead to overfitting of the data. Hence, one of the ways to prevent overfitting is to limit complexity of neural network by using optimal number of hidden layers and number of nodes in hidden layer.

```
Best score:0.8479
Best parameter:  {'hidden_layer_sizes': 4}  *
```

**Note*: The same block of code produced different results each time it was run. This was the output received while writing the report.

Mean and standard deviation score for each hidden layer is as shown below with highest hidden score of 4 highlighted in red.

	param_hidden_layer_sizes	mean_test_score	std_test_score
0	2	0.818405	0.003006
1	3	0.818405	0.003006
2	4	0.847853	0.016644
3	5	0.833129	0.014203
4	6	0.833129	0.011896
5	7	0.839264	0.016185
6	8	0.835583	0.022086
7	9	0.833129	0.015227
8	10	0.831902	0.015809
9	11	0.844172	0.013773
10	12	0.833129	0.014203
11	13	0.839264	0.015227
12	14	0.844172	0.017611
13	15	0.833129	0.021742
14	16	0.833129	0.018364
15	17	0.834356	0.020895
16	18	0.830675	0.024723
17	19	0.826994	0.008139



By training neural network using MLPClassifier() with best identified value of the parameter of hidden layer node as 4. Below are the final intercepts and network weights of the improved neural network model.

```
Final Intercepts for Red Wine Quality Neural Network Model
[array([0.13422681, 0.55361136, 0.56849142, 0.05948397]), array([-1.06076433,  0.03216902,  0.43383568])]

Network Weights for Red Wine Quality Neural Network Model
[array([[-0.26172065,  0.14968688, -0.22545135,  0.07254314],
       [ 1.67084795,  2.81209955,  0.98346332,  0.04677778],
       [-2.09298788, -2.29944618, -0.56108765,  0.37425744],
       [-0.53397653, -0.03042432,  0.31473206,  0.30844603],
       [ 1.19853059,  1.80865582,  0.33929426, -0.00321491],
       [-0.18886402, -0.03811498, -0.05011699,  0.37707165],
       [ 0.14325678,  0.01072338, -0.34687728, -0.03391461],
       [ 0.22007074,  0.85997231,  0.10857694,  0.15123133],
       [ 0.76459692,  0.65369956,  0.50585695,  0.07092908],
       [-1.12137064, -1.33555912,  0.28862463,  0.39944221],
       [-0.08750264, -0.44494622,  0.26094212,  0.41612003]]), array([[-0.26710227,  0.24476658, -1.71152242],
       [ 0.53160701,  0.45911319, -2.04419715],
       [ 1.38345845, -1.04975397, -0.29452057],
       [-1.43893564,  0.47548524,  0.32297002]])]
```

The first array of the Final Intercepts contains node bias values θ for 4 nodes in the hidden layer. The second array of the Final Intercepts contains the θ values for the three nodes in the output layer.

The first array of the Network Weights contains weights from each of the 11 nodes in the input layer (11 predictors' nodes) to the 4 nodes in the hidden layer. The second array of the Network

Weights contains weights from each of the 4 nodes in the hidden layer to the three node in the output layer.

The table with the actual and classification results and associated probabilities for the first 10 records in the validation partition are presented below.

Classification for Red Wine Quality Data for Validation Partition

	Actual	p(0)	p(1)	p(2)	Classification
473	1	0.0470	0.5851	0.3679	1
1376	1	0.0309	0.9466	0.0225	1
533	1	0.0288	0.6645	0.3067	1
200	2	0.0314	0.7110	0.2576	1
268	1	0.0329	0.9260	0.0411	1
210	1	0.0339	0.6124	0.3537	1
906	1	0.0287	0.9290	0.0423	1
1093	2	0.0290	0.7022	0.2688	1
947	2	0.0747	0.4409	0.4844	2
867	1	0.0346	0.7407	0.2248	1

The majority of records are classified correctly as 1 (‘medium). The 2 out of 10 records with indexes of 200, 1093 are misclassified. They have the actual quality_category status as 2 (‘high), but the neural network model misclassified them as 1 (‘medium’).

Classification Summary for Training Partition using Neural Network Model
Confusion Matrix (Accuracy 0.8209)

	Prediction		
Actual	0	1	2
0	0	32	1
1	0	661	6
2	0	107	8

Classification Summary for Validation Partition using Neural Network Model
Confusion Matrix (Accuracy 0.8217)

	Prediction		
Actual	0	1	2
0	0	30	0
1	0	442	3
2	0	64	5

The confusion matrices for the training and validation partitions show a very good accuracy of close to 82%, and thus the trained neural network model fits well for the validation date set and can be used for classification of the red wine quality testing. The misclassification rate for the

training partition is 1-0.8209= 0.1791 or 17.91%, and for the validation partition 1-0.8217 = 0.1783 or 17.83%. The accuracy of the model for the validation records is even slightly higher that the accuracy for the training records, and therefore, there is no overfitting in this case.

Step 9 – Interpreting Results

A closer look shows that all the models are struggling particularly in predicting those wines that are low in quality – they are unable to predict even one record correctly. Similarly with high-quality wines the models seem to struggle – less than half high-quality wines were correctly predicted. A possible reason could be if the number of records for high- and low-quality wines was less, the model could not have had the opportunity to sufficiently "learn".

Confusion matrix for validation partition seems to also struggle with correctly predicting high- and low-quality wines. Its performance in validation partition is similar to that in training partition – less than half high-quality wines were correctly predicted, and none of the low-quality wines were correctly classified.

```
Classification Summary for Validation Partition using Classification Tree
Confusion Matrix (Accuracy 0.8382)

       Prediction
Actual   0   1   2
     0  26   0  43
     1   0   0  30
     2  15   0 430

Classification Summary for Validation Partition using Neural Network Model
Confusion Matrix (Accuracy 0.8217)

       Prediction
Actual   0   1   2
     0   0  30   0
     1   0 442   3
     2   0  64   5

Accuracy Measure for Validation Partition for Ordinal Logistic Model
Confusion Matrix (Accuracy 0.8180)

       Prediction
Actual   0   1   2
     0   1  28   1
     1   0 423  22
     2   0  48  21
```

For the three confusion matrices provided above, the highest accuracy of 0.8382 or 83.82% (misclassification rate of 16.18%) was achieved in the Classification Tree model, and therefore, this model would be recommended for the classification of the red wine quality.

Step 10 – Deploy the best technique

The Classification Tree model was used to classify three new data records, the results of which are below.

Classifications for Red Wine Quality

	fixed_acidity	volatile_acidity	citric_acid	residual_sugar	chlorides	free_sulfur_dioxide	total_sulfur_dioxide	density	pH	sulphates	alcohol	classification
0	7.2	0.55	0.25	1.8	0.083	28	93	0.9966	3.29	0.56	10.8	medium
1	6.2	0.66	0.32	1.2	0.076	27	92	0.8977	3.15	0.46	10.7	medium
2	6.7	0.45	0.45	2.6	0.072	26	80	0.9060	2.99	0.70	11.6	high

Conclusion

All three models performed reasonably well in predicting the quality of wine. However, classification tree had the highest accuracy at 83%, followed closely by neural network model and then by ordinal logistic regression model. Overall, the results of the project suggest that any of the three models could be used to predict the quality of red wine based on predictor variables.

Additionally, all models are confidently recommended for predicting medium quality wines, but only cautiously recommended for low- and high-quality wines. In order to strengthen the model for predicting low- and high-quality wines, the authors recommend training the model with increased data for low- and high-quality wines.

Bibliography

Cortez, P., Cerdeira, A., Almeida, F., Matos, T., & Reis, J. (2009). Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, *47*(4), 547–553. https://doi.org/10.1016/j.dss.2009.05.016

Red Wine Quality. (n.d.). Retrieved May 12, 2023, from https://www.kaggle.com/datasets/uciml/red-wine-quality-cortez-et-al-2009