Wine Taste Preference Modeling Based On Physicochemical Tests

Shuai Wei

April 25, 2016

Introduction

Wine, as a social drinking, is enjoyed by more and more consumers nowadays. Wine quality assessment is the key part in exploring new technologies for the wine making and selling process. The important aspect in wine quality assessment is physicochemical tests which characterize the factors such as alcohol, acidity and sulphates. The price of wine relies on a quite conceptual feeling of wine tasters, and opinions may vary in a great range. Thus, modeling the wine taste preference based on physicochemical tests is always a tough and significant task in wine industry. After regarding the discrete response as a continuous variable, we are going to have a regression problem. The main purpose of this study is to explore the complex relationship between physicochemical properties and taseter's rating by trying different kinds of regression models.

Data

Data collection

A total of 4898 white and 1599 red wine samples are considered, related to red and white variants of the Portuguese “Vinho Verde” wine, which is known as “Green Win”. All wine data is from a particular area of Portugal and it is collected on 11 chemical properties and 1 sensory data “quality”.

The 11 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 is quality, which is ranging from 1 to 9 and we just consider it as a continuous variable to do regression analysis.

Exploratory Data Analysis

The following is partial variables summary result for white wine data.

Table 1: Summary of 4 variable in white wine data

fixed.acidity	volatile.acidity	citric.acid	residual.sugar
Min. : 3.800	Min. :0.0800	Min. :0.0000	Min. : 0.600
Median : 6.800	Median :0.2600	Median :0.3200	Median : 5.200
Mean : 6.855	Mean :0.2782	Mean :0.3342	Mean : 6.391
Max. :14.200	Max. :1.1000	Max. :1.6600	Max. :65.800

From the table, we can see that for input residual.sugar, the mean and median is a little different compared with other inputs and the maximal value 65.8 is far away from the mean 6.391 while the minimal 0.6 is much closer to the mean. Hence, we can assume that there are outliers in the data set. Then based on boxplox rule, we regard a predictor value as outlier only if it is not in the range $[Q_1 - c * IQD, Q_3 + c * IQD]$, where Q_1 and Q_3 represent the lower and upper quartiles, respectively and $IQD = Q_3 - Q_1$ is the interquartile distance. The threshold parameter c commonly used is 1.5, however, to keep more data, we just make c be 3.

Application of this rule reduces the white wine data size from 4898 to 4690, and reduces the red whine data size from 1599 to 1435.

After removing all the outliers, we get the histogram plot of quality for wine data.

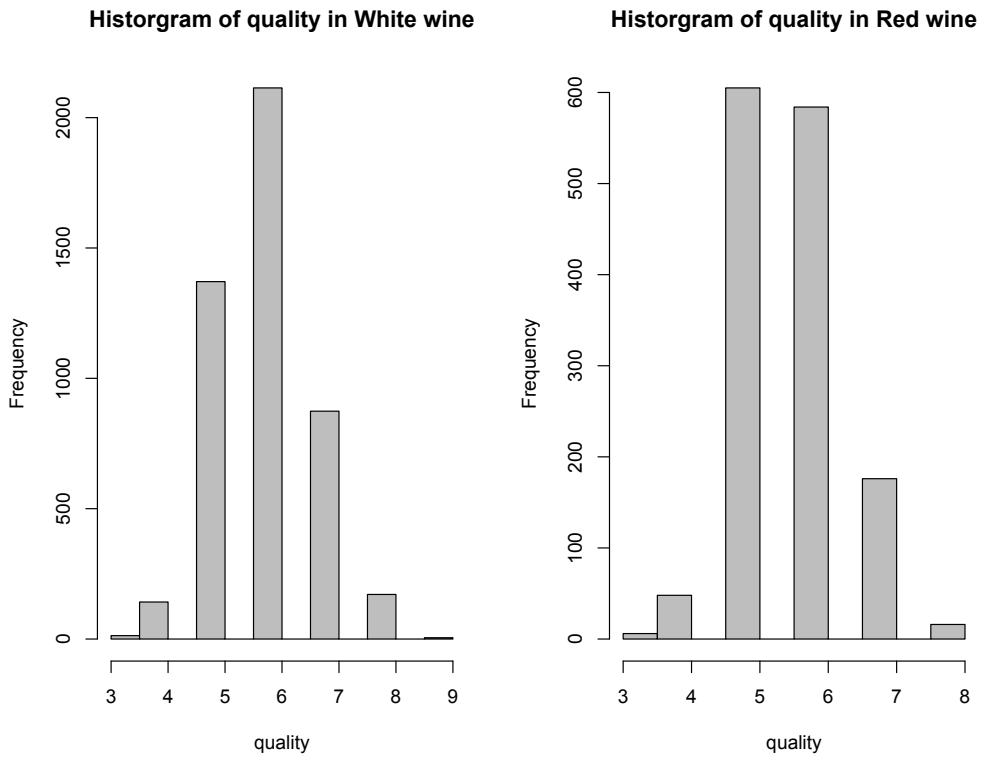

Figure 1: Histogram of quality of wine data

It looks like a normal distributed shape from the histogram for the two kinds of data sets

and there are much more normal wines than excellent and poor ones.

Afterwards, we create a new predictor whiteOrRed for the two kinds of wines, whose value is 1 for white wine data and 0 for red one. Then we combine the two data sets into one in order to make our analysis become simple and convenient. Now the number of predictors becomes 12.

At last, in order to verify our model's performance for unknown data, we split data randomly into two groups which will be used for all of our models. One group with size 2/3 of the data is used for training and the other group for testing.

Statistical learning

Multiple Linear regression

At first, we put all predictors into our multiple linear regression model, and we get VIF for each predictor.

Table 2: VIF table

fixed.acidity	volatile.acidity	citric.acid	residual.sugar	chlorides	free.sulfur.dioxide
5.6646	2.3152	1.6819	12.3840	3.4784	2.2515
total.sulfur.dioxide	density	pH	sulphates	alcohol	whiteOrRed
4.2032	31.0751	2.7204	1.5505	7.5554	9.6113

So we ignore predictors density and residual.sugar from the model in multiple linear regression since they have a pretty high VIF which is more than 10, and there is high probability that both of them have collinearity with other predictors.

After excluding variables density and residual.sugar, we fit a updated multiple linear regression model. The primary summary results are showed below.

Table 3: Partial Coefficient table

	Estimate	Std. Error	t value	Pr(>|t|)
fixed.acidity	0.0203	0.0140	1.45	0.1467
volatile.acidity	-1.4049	0.1124	-12.50	0.0000
citric.acid	-0.0403	0.1096	-0.37	0.7130
residual.sugar	0.0211	0.0031	6.91	0.0000

According to the p-value from the above table, we know not all predictors are statistically

significant. Therefore, variable selection methods are needed before we implement our regression analysis.

A forward selection method is used to build a preliminary regression model, which just chooses 5 relatively useful predictors. They are volatile.acidity, free.sulfur.dioxide, total.sulfur.dioxide, sulphates and alcohol. By the way, backward selection and best subset selection method choose totally the same subsets of predictors.

Based on the 5 variables selected, we implement our preliminary multiple linear regression analysis, and we have the following statistical summary results.

Table 4: Coefficient table

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	2.5448	0.1437	17.71	0.0000
volatile.acidity	-1.3614	0.0847	-16.07	0.0000
free.sulfur.dioxide	0.0088	0.0010	8.94	0.0000
total.sulfur.dioxide	-0.0019	0.0003	-5.73	0.0000
sulphates	0.6919	0.0920	7.52	0.0000
alcohol	0.3175	0.0103	30.87	0.0000

Residual standard error: 0.7393 on 4077 degrees of freedom

Multiple R-squared: 0.2794, Adjusted R-squared: 0.2785

F-statistic: 316.1 on 5 and 4077 DF, p-value: $<$ 2.2e-16

According to the analysis results, it appears that all the 5 predictors chosen are statistically significant. Nevertheless, the multiple R^2 is just 27.94%. Thus we would like to adjust and optimize the model.

Then we plot the residual diagnostics of each predictor except the dummy variable whiteOrRed1 versus quality. From component residual plot not showing here, we know obvious polynomial trend exists for pairs of predictors and the response. Then we try to add some quadratic and cubic terms to the preliminary model.

Furthermore, we attempt to add some interaction terms and get a better multiple linear regression model. Finally, we get our final multiple linear regression model, in which each coefficient is statistically significant and the R^2 is 30.42. It is showed below.

$$\begin{aligned} quality = {} & (total.sulfur.dioxide) + (volatile.acidity) + (free.sulfur.dioxide) \\ & + (free.sulfur.dioxide)^2 + sulphates + alcohol + alcohol^2 \\ & + alcohol^3 + (volatile.acidity) * alcohol + (free.sulfur.dioxide) \\ & * sulphates + (free.sulfur.dioxide) * alcohol \end{aligned} \tag{1}$$

According to the model we get the training MSE 0.5270 based on the 2/3 of training data set. To our surprise, the testing error for the 1/3 of testing one is 0.5237, and it seems we fit a good multiple linear regression model.

Lasso Method

The lasso method has a good property of shrinking the coefficient not significant estimates towards zero compared with ridge regression, and the advantage of variable selection of it is pretty suitable for our modeling question since we want our model as simple as possible.

After applying lasso method, the coefficients for 8 variables chosen are shown as follows.

Table 5: Coefficient table

X.Intercept.	volatile.acidity	residual.sugar	chlorides	free.sulfur.dioxide
2.2768	-1.2805	0.0180	-1.2799	0.0057
total.sulfur.dioxide	pH	sulphates	alcohol	whiteOrRed1
-0.0012	0.0477	0.6560	0.3322	-0.0882

The best λ chosen by cross validation is 2.7470×10^{-4}. Compared with the preliminary multiple linear regression model without polynomial and interaction terms, they have very similar coefficients except that it just uses 4 more predictors from the original data. Besides, the positive or negative linear relationship between each predictor and the response quality is consistent based on the sign of each fitting coefficient. It makes sense since the best λ is very close to 0.

The test MSE of lasso method is 0.5148.

Regression Tree

Regression tree involves partitioning the data space into several simple regions. We use the mean in the region to which a observation belongs to as the its prediction value. It is simple and has compelling interpretability. Despite that, in terms of prediction accuracy, it is less competitive compared to some excellent supervised learning method.

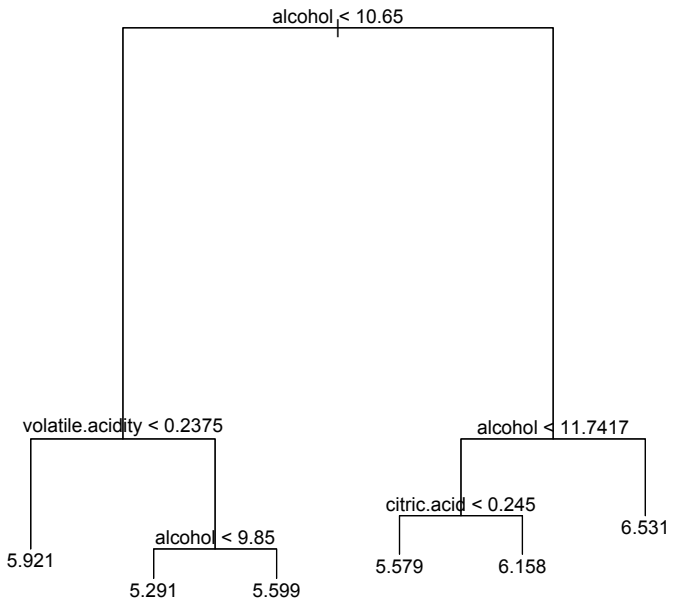

Figure 2: The regression tree

In the regression tree model, just 3 predictors are used, which are alcohol, volatile.acidity and citric.acid. There are 6 terminal nodes in the tree in total.

From the regression tree plot, we can see that wine with less alcohol has worse quality and when the variable alcohol is fixed, wine with less volatile.acidity has better quality. It makes sense because alcohol is one of the most important component in wine and people prefer the wine tasting sweeter with less acidity.

The test MSE of the method of regression tree is 0.5470.

Random Forests

Random forests also belong to the tree-based method which can reduce the variance compared to regression tree. They are especially useful and often applied in the background of decision trees. Random forests have an improvement over bagged trees by decorrelating the trees. At each split in the tree, random forests just consider only a subset of the predictors in order to reduce the possibility of existing highly correlated bagged trees. It seems that they can be used in our model in the view of the fact that there is collinearity among predictors.

By the variables importance plot not showing here, the two most important predictors are

alcohol and volatile.acidity with IncMSE 73.58 and 67.66, respectively. The least two important predictors are density with IncMSE 34.9882 and the new added predictor whiteOrRed 10.91. It is consistent with regression tree which also chooses these two important predictors. Moreover, whiteOrRed is the least important, so our idea of considering the two data sets as one is not so bad as far as random forests are considered.

The number of predictors are 12, so after choose $m = 3$, we have the test MSE for random forests is 0.3580.

Performance Comparison

The test MSE of each method is displayed as follows.

Table 6: Test MSE of each method

Multiple linear regression	Lasso	Regression tree	random forests
0.5237	0.5148	0.5470	0.3580

When considering test MSE, the rank of these methods are: Random forests > Lasso > Multiple Regression > Regression tree.

The regression tree method suffers higher variance itself, that is, the results may be quite different when training data is split into two groups randomly. So we first exclude regression tree method since it also has the biggest test MSE.

Although the linear regression tends to have lower variance compared with regression tree given the ratio of the size of data n to the number of predictors p is large enough, in variable selection process, it is not stable because the selection results differ very much when the size of training set varies.

It seems that we should choose random forests since they have pretty low test MSE, but as flexible methods, they are less interpretable and have much higher variance compared with lasso method. Moreover, in our wine analysis problem, we need to make our model more stable and more interpretable so that we can find the accurate relationship between each predictor and the response such that the model is more understandable and applicable to the real life.

In Best Model

In our lasso model, we find that volatile.acidity, residual.sugar, chlorides and total.sulfur.dioxide has a negative linear relationship with the response quality, and residual.sugar, free.sulfur.dioxide, pH, sulphates, alcohol has a positive relationship with the quality. It looks reasonable, by the reason of that if there is more acidity and sulfur, it is less likely that higher rating is given, and the more there is alcohol and sugar, the more likely people like it. In summary, if we add a little more sugar or alcohol into wine while controlling the amount of acidity and sulfur, the wine made could be preferred by much more consumers.

Conclusion

In preprocessing our data, we combine two wine data into one set and based on it we conduct our data analysis. It seems unreasonable since in most cases we may build different models for different kind wines, but to our surprise, all the 5 models we tried produce a pretty good analysis results. In our best model, the dummy variable whiteOrRed1 is not included in the model, giving evidence that same model can be applied to the white wine and red wine sample.

In the process of finding best model, we find alcohol and volatile.acidity are the two key factors for our data based on random forests model. Thus, if the wine manufacturer can pay attention to that in making wine, they may produce more satisfactory red or white wine. One shortcoming of our model is that when calculating the test MSE, we do not round the fitted value to an integer close it. If that is done, we should have had a more precise and more significant test MSE. What's more, not like multiple linear regression method, we do not test the significance of the predictor variable that enters the current lasso model. This is the part we should consider in the future.

References

[1] 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.

R Code

```
#input wine data and
WhiteWine=read.csv('white.csv', head=T, row.names = NULL, sep=';')
WhiteWine = na.omit(WhiteWine)

#output part White Wine result min, mean, median and max to document
library(xtable)
xtable(summary(WhiteWine)[c(1,3,4,6),1:4])

RedWine = read.csv('red.csv', head=T, row.names = NULL, sep=';')
RedWine = na.omit(RedWine)

# remove outliers for White Wine if > Q3 + 3IQR or < Q1    3IQR
rowsKept=rep(TRUE, dim(WhiteWine)[1])
for (col in names(WhiteWine)){
        data = WhiteWine[, col]
        iqr = IQR(data)
        lowerq = quantile(data)[2]
        upperq = quantile(data)[4]
        mild.threshold.upper = (iqr * 3) + upperq
        mild.threshold.lower = lowerq    (iqr * 3)
        rowsKept = rowsKept & ((WhiteWine[,col] <= mild.threshold.upper)
        & (WhiteWine[,col] >= mild.threshold.lower))
}
WhiteWine = WhiteWine[rowsKept, ]
summary(WhiteWine)


# remove outliers for Red Wine if > Q3 + 3IQR or < Q1    3IQR
```

```
rowsKept=rep(TRUE, dim(RedWine)[1])
for (col in names(RedWine)){
        data = RedWine[, col]
    iqr = IQR(data)
        lowerq = quantile(data)[2]
        upperq = quantile(data)[4]
        mild.threshold.upper = (iqr * 3) + upperq
        mild.threshold.lower = lowerq   (iqr * 3)
        rowsKept = rowsKept & ((RedWine[,col] <= mild.threshold.upper)
        & (RedWine[,col] >= mild.threshold.lower))
}
RedWine = RedWine[rowsKept, ]
summary(RedWine)

par(mfrow = c(1,2))
h = hist(WhiteWine$quality, col= "grey",xlab="quality",
                    main = "Historgram of quality in White wine")
h = hist(RedWine$quality, col= "grey", xlab="quality",
                    main = "Historgram of quality in White wine")

#To analyze white and red wine together, we remove the density and
#residual.sugar variables

#combine two data sets into one.
WhiteWine["whiteOrRed"] = (rep(1, dim(WhiteWine)[1]))
RedWine["whiteOrRed"] = (rep(0, dim(RedWine)[1]))
Wine = rbind(WhiteWine, RedWine)
Wine$whiteOrRed = as.factor(Wine$whiteOrRed)

#Generate train and test set
```

```
set.seed(2016) # do not forget this.
train=sample(1:nrow(Wine), nrow(Wine)*2/3)
test=(train)
trainWine = Wine[train,]

#remove variable which has large vif
lm.fit= lm(quality~., trainWine)
library(car)
vif(lm.fit)
#output vif table to document here.

#Multiple linear regression
lm.fit = lm(quality~. density, data = trainWine)
xtable(summary(lm.fit))

#forward variable selection
library(leaps)
regfit.fwd=regsubsets(quality~. density residual.sugar, trainWine,
                      nvmax = 12,method="forward")
regfit.summary = summary(regfit.fwd)
which.min(regfit.summary$bic)
coef(regfit.fwd, 5)

#cv for variables selection
# 10 folds cv
Wine1 = trainWine[,c(1,2,3,5,6,7,9,10,11,12,13)]

predict.regsubsets=function(object,newdata,id,...){
            form=as.formula(object$call[[2]])
        mat=model.matrix(form,newdata)
```

```
    coefi=coef(object,id=id)
    xvars=names(coefi)
        mat[,xvars]%*%coefi
}

k=10
set.seed(2016)
folds=sample(1:k,nrow(Wine1),replace=TRUE)
varNums = 10
cv.errors=matrix(NA,k,varNums, dimnames=list(NULL, paste(1:varNums)))

for(j in 1:k){
    best.fit=regsubsets(quality~.,data=Wine1[folds!=j,],nvmax=varNums)
    for(i in 1:varNums){
        pred=predict(best.fit,Wine1[folds==j,],id=i)
        cv.errors[j,i]=mean( (Wine1$quality[folds==j] pred)^2)
    }
}
mean.cv.errors=apply(cv.errors,2,mean)
mean.cv.errors
par(mfrow=c(1,1))
plot(mean.cv.errors,type='b')
reg.best=regsubsets(quality~.,data=Wine1, nvmax=varNums)
coef(reg.best,7)

#The left variables are: volatile.acidity,free.sulfur.dioxide,
#sulphates, alcohol, total.sulfur.dioxide

lm.fit=lm(quality~., data = trainWine[,c(2,6,7,10,11,12)])
summary(lm.fit)
```

```
#check if interaction and polynomial trend exist
library(car)
crPlots(lm.fit, layout= c(2,3))
##add all interation

#add polynomial and interaction
lm1.fit=lm(quality~ total.sulfur.dioxide
                    + volatile.acidity
                    + free.sulfur.dioxide +I(free.sulfur.dioxide^2)
                    + sulphates
                    + alcohol + I(alcohol^2) + I(alcohol^3)
                    + volatile.acidity:alcohol
                    + free.sulfur.dioxide:sulphates
                    + free.sulfur.dioxide:alcohol
                    ,data = trainWine[,c(2,6,7,10,11,12)])

#calculate train MSE
mean(lm1.fit$residuals^2)
#calculate the test MSE
mlr.pred=predict(lm.fit, Wine[test,c(2,6,7,10,11)])
y.test = Wine[test,12]
mean((mlr.pred y.test)^2)


####lasso

x=model.matrix(quality~.,Wine)[, 1]
y=Wine$quality
library(glmnet)
```

```
grid=10^seq(10, 2, length=100)
y.test=y[test]


lasso.mod=glmnet(x[train,],y[train],alpha=1,lambda=grid)
plot(lasso.mod)
set.seed(2016)
#default nfolds is 10
cv.out=cv.glmnet(x[train,],y[train],alpha=1,nfolds=10)
plot(cv.out)
bestlam=cv.out$lambda.min
lasso.pred=predict(lasso.mod,s=bestlam,newx=x[test,])
mean((lasso.pred y.test)^2)
out=glmnet(x,y,alpha=1,lambda=grid)
#13 is the total numbers of variables !!
lasso.coef=predict(out,type="coefficients",s=bestlam)[1:13,]
lasso.coef
lasso.coef[lasso.coef!=0]

######
#regression tree
library(tree)
tree.carseats=tree(quality~., Wine,subset=train)
summary(tree.carseats)
plot(tree.carseats)
text(tree.carseats,pretty=0,cex=1)

yhat=predict(tree.carseats,newdata=Wine[ train,])
carseats.test=Wine[ train,"quality"]
#MSE here to get result
```

```
MSE1 = mean((yhat - carseats.test)^2)
set.seed(2016)
cv.carseats=cv.tree(tree.carseats)
plot(cv.carseats$size, cv.carseats$dev, type = "b")

tree.min < which.min(cv.carseats$dev)



#bagging approach

library(randomForest)
#random forests
# B/3
set.seed(2016)
rf.carseats=randomForest(quality~.,data=Wine,subset=train,mtry = 3,
                         importance=TRUE)
yhat.rf = predict(rf.carseats,newdata=Wine[-train,])
MSE4=mean((yhat.rf - carseats.test)^2)
importance(rf.carseats)
```