Project 3: FINAL PROJECT

Math 6375 - 2022 Spring

Achraf Cherkaoui

Red Wine Quality:

We consider the wine dataset.

Data description :

For this study I will analyze a Red Wine data set created by:
Paulo Cortez (Univ. Minho), Antonio Cerdeira, Fernando Almeida, Telmo Matos and Jose Reis (CVRVV) @ 2009. This data set has 1599 observation and it contains 11 input variables: fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol and the output variable quality.

Input variables (based on physicochemical tests):all the variables are numerical.

1 - Fixed Acidity: Most acids involved with wine or fixed or nonvolatile (do not evaporate readily) (tartaric acid - g/dmˆ3)

2 - Volatile Acidity: The amount of acetic acid in wine, which at too high of levels can lead to an unpleasant, vinegar taste. (acetic acid - g/dmˆ3)

3 - Citric Acid: Found in small quantities, citric acid can add 'freshness' and flavor to wines. (g/dmˆ3)

4 - Residual Sugar: The amount of sugar remaining after fermentation stops, it's rare to find wines with less than 1 gram/liter and wines with greater than 45 grams/liter are considered sweet. (g/dmˆ3)

5 - Chlorides: The amount of salt in the wine. (sodium chloride - g/dmˆ3)

6 - Free Sulfur Dioxide: The free form of SO2 exists in equilibrium between molecular SO2 (as a dissolved gas) and bisulfite ion; it prevents microbial growth and the oxidation of wine. (mg/dmˆ3)

7 - Total Sulfur Dioxide: Amount of free and bound forms of S02; in low concentrations, SO2 is mostly undetectable in wine, but at free SO2 concentrations over 50 ppm, SO2 becomes evident in the nose and taste of wine. (mg/dmˆ3)

8 - Density: The density of water is close to that of water depending on the percent alcohol and sugar content. (g/cmˆ3)

9 - pH: Describes how acidic or basic a wine is on a scale from 0 (very acidic) to 14 (very basic); most wines are between 3-4 on the pH scale

10 - Sulphates: A wine additive which can contribute to sulfur dioxide gas (S02) levels, wich acts as an antimicrobial and antioxidant. (g/dm3)

11 - Alcohol: The percent alcohol content of the wine

Output variable (based on sensory data):

12 - quality (score between 0 and 10)

GOAL :

we are going to use machine learning to determine which physiochemical properties make wine 'good'! Three methods are compared for this data set, the best one is selected and applied to the data for the final model in order to determine the quality of wine.

METHOD :

we are going to apply three different supervised learning algorithms on this data,Multiple linear regression(MLR),neural network(NN) and principal component regression(PCR). using 10-fold cross validation to compare the performance of the three methods we are going to choose the best method that give us the best prediction model for the wine quality.

First we split the data into training and test set. The training set should be used to build our machine learning models. in other words we are going to create a model based on our training data then we test our model on the test data.The test set should be used to see how well our model performs on unseen data.

The quality is the univariate response y. we need a couple of steps:

- Step 1. Construct 2 nested for loops, where the outer for loop is for 10-fold cross validation in order to compare the performance of the three methods; the inner for loop is for finding the best tuning parameter value for the NN model:
  - Within the outer for loop, the MLR,NN and PRC are fit on the train set, predictions on the test set are made, and test errors (MSE) are obtained. The model fitting, prediction on test set, and calculation of test error given by NN must be done after the inner for loop determines the best number of nodes.
- Step 2. Make boxplot for the 10 cross-validation errors of the three methods. Perform paired t test for significance of difference of CV errors between the three methods.
- Step 3. Choose the one with better performance, fit to the whole data, to get the final model.

Multiple linear regression:

Multiple linear regression (MLR), is a statistical technique that uses several explanatory variables to predict the outcome of a response variable. The goal of multiple linear regression is to model the linear relationship between the explanatory (independent) variables and response (dependent) variables.

Advantages of Multiple linear Regression: There are two main advantages to analyzing data using a multiple regression model. The first is the ability to determine the relative influence of one or more predictor variables to the criterion value.The second advantage is the ability to identify outliers, or anomalies.

Disadvantages of Multiple linear Regression -Linear Regression Only Looks at the Mean of the Dependent Variable. -Linear regression looks at a relationship between the mean of the dependent variable and the independent variables. -Linear Regression Is Sensitive to Outliers.also the Data Must be Independent.

Neural Network :

A neural network is a series of algorithms that endeavors to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates. In this sense, neural networks refer to systems of neurons, either organic or artificial in nature

Advantages of Neural Network -Store information on the entire network. -The network problem does not immediately corrode. -Ability to train machine: Artificial neural networks learn events and make decisions by commenting on similar events. -Parallel processing ability: Artificial neural networks have numerical strength that can perform more than one job at the same time.

Disadvantages of Neural Network Neural networks usually require much more data than traditional machine learning algorithms, as in at least thousands if not millions of labeled samples. This isn't an easy

problem to deal with and many machine learning problems can be solved well with less data if you use other algorithms.

Principal component regression:

principal component regression (PCR) is a regression analysis technique that is based on principal component analysis (PCA). More specifically, PCR is used for estimating the unknown regression coefficients in a standard linear regression model.

Advantages of PCR -Removes Correlated Features. -Improves Algorithm Performance. -Reduces Overfitting -Improves Visualization:

Advantages of PCR - Independent variables become less interpretable. - Data standardization is must before PCR. - Information Loss: Although Principal Components try to cover maximum variance among the features in a dataset, if we don't select the number of Principal Components with care, it may miss some information as compared to the original list of features.

```
library(nnet)
library(MASS)
library(pROC)
library(randomForest)
library(ggplot2)
library(ElemStatLearn)
library(pls)
library("openxlsx")
library(MASS)
library(ggplot2)
library(readxl)
library(tidyverse)
wine <- read_excel("winequality-red.xlsx")
sum(is.na(wine)) # no missing values in the data
```

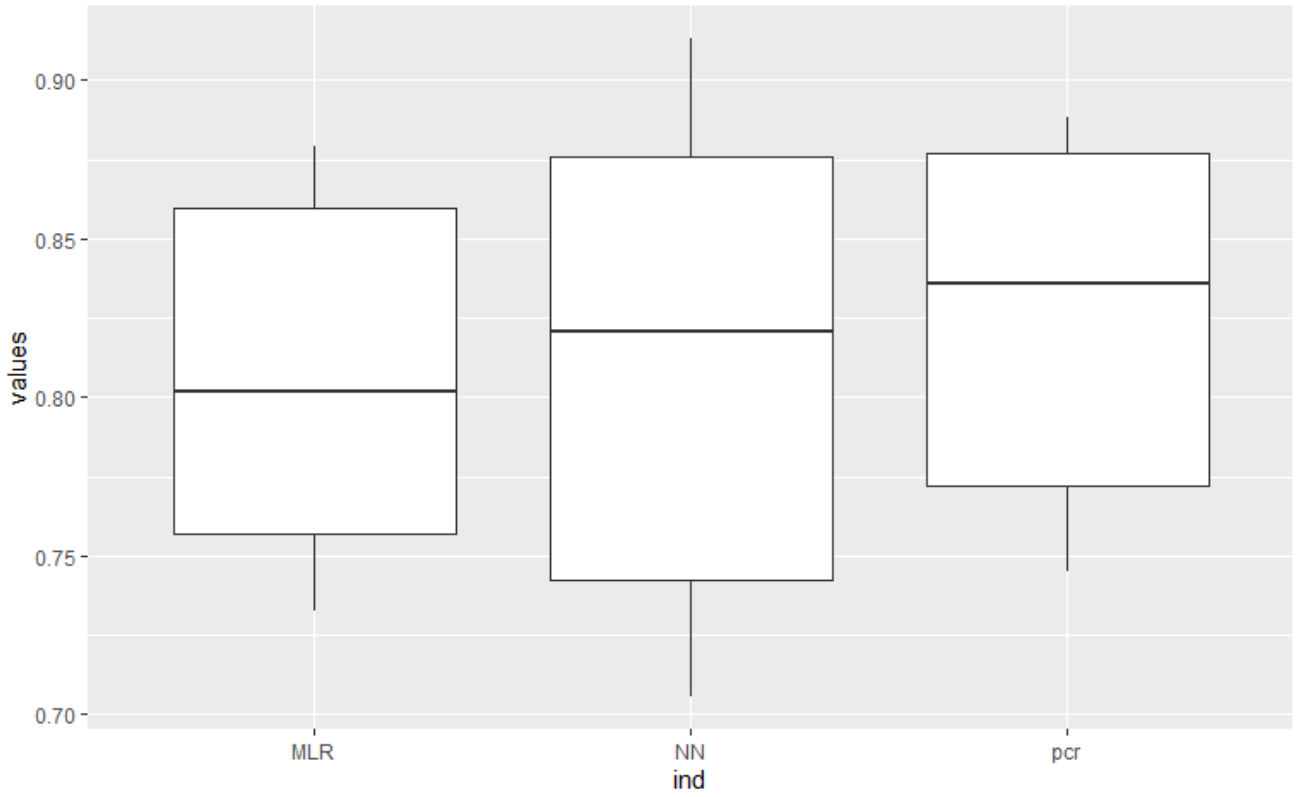
```
## [1] 0
```

```
library(nnet)
# Set K=10 to perform 10-fold cross validation
K =10
set.seed(123) # always set seed before randomization

# Randomly divide and then assign all data points to the K folds.
fold.assignments =sample(rep(1:K,length=nrow(wine)))

# Initialize the err.cv matrix to store the 10 cross-validation errors for LM and NN.
# It is a KX2 matrix, the first column is for MLR, the 2nd is for NN.
err.cv = matrix(0,K,3)
colnames(err.cv) = c("MLR","NN","pcr")

# The outer "for" loop
# The loop is to perform cross-validation for the two methods using the same data every time.
for (k in 1:K){
  # Print out progress
  cat("Fold",k,"... ")

  # Partition into training and test sets
  inds = which(fold.assignments==k)
  test = wine[inds, ]# data in the kth fold is the test set
```

```r
train = wine[-inds, ] # the remaining is for the train set



# set up grid for the values of tuning parameters.
grid=c(5,10,15,20,25,30,50)

M = 10 # 10-folds CV for inner loops
set.seed(1)
inner.fold.assignments =sample(rep(1:M,length=nrow(train)))

NN.err.cv = matrix(0,length(grid),M)
rownames(NN.err.cv) = grid
colnames(NN.err.cv) = paste("fold_",1:M,sep="")

# inner for loop
for (j in 1:M){
  inner.inds = which(inner.fold.assignments==j)

  inner.train = train[-inner.inds,]

  mean = apply(inner.train,2,mean)
  std = apply(inner.train,2,sd)

  new = scale (train,center=mean,scale=std)

  new.train = new[-inner.inds,]
  new.test =new[inner.inds,]

  for (i in 1:length(grid)){
    num_nodes= grid[i]

    fit = nnet(quality~.,data=new.train,size=num_nodes,
               linout=TRUE,maxit=200,trace=FALSE)
    pred = predict(fit,newdata=new.test)
    NN.err.cv[i,j] = mean((pred[,1]-new.test[,"quality"])^2)
  }
}
mse = rowMeans(NN.err.cv)

# the best number of nodes
outer.num_nodes = as.numeric(names(mse)[which.min(mse)])


#(a) Next, we normalize the train and test set using info of train set.
# Hint: refer to lines 4-8 of Adv_StatLearning_4_part_3_annotated.pdf.
mean.train<- apply(train, 2, mean)
std.train <- apply(train, 2, sd)
train_data <- as.data.frame(scale(train, center = mean.train, scale = std.train))
test_data <- as.data.frame(scale(test, center = mean.train, scale = std.train))

x_train = model.matrix(quality~., train_data)[,-1]
x_test = model.matrix(quality~., test_data)[,-1]
```

```
  y_train = train_data %>%
  select(quality) %>%
  unlist() %>%
  as.numeric()

  y_test = test_data %>%
  select(quality) %>%
  unlist() %>%
  as.numeric()

  #(b) Then, fit lm() ,nnet() and PCR to the normalized train set, where nnet() must
  #include the argument: size = outer.num_nodes.
  nn_model = nnet(quality~., data=train_data, size = outer.num_nodes,
                  linout=TRUE, maxit=200, trace=FALSE)
  lm_model = lm(quality~., data=train_data)
  pcr_fit2 = pcr(quality~., data = train_data, scale = TRUE, validation = "CV")

  #(c) Then, make predicttion on the normalized test set.
  nn.prediction = predict(nn_model, newdata = test_data, type="raw")
  lm.prediction = predict(lm_model, newdata = test_data)
   bestM = selectNcomp(pcr_fit2, "onesigma", plot = TRUE)
  tuning.best [k,3 ] =bestM
  pcr_pred = predict(pcr_fit2, x_test, ncomp=bestM)

  #(d) Calculate the mean square error for the normalized test sets and store
  #them to the kth row of the matrix: err.cv.
  err.cv[k, 1] = sqrt(mean((lm.prediction-test_data$quality)^2))
  err.cv[k, 2] = sqrt(mean((nn.prediction-test_data$quality)^2))
  err.cv[k, 3] = sqrt(mean((pcr_pred-y_test)^2))


  ##validationplot(pcr_fit2, val.type = "MSEP")
  #validationplot(pcr_fit2, val.type = "R2")

}
#(e) Please draw a boxplot for err.cv for the test errors for the MLR and NN models.
# If you use ggplot(), you might need the melt() in `reshape` package to reshape
# the err.vc dataframe.
err.cv.df <- stack(as.data.frame(err.cv))
Bplot <- ggplot(data=err.cv.df)+geom_boxplot(aes(x=ind, y=values))
Bplot


#(f) Perform paired t-test using t.test() to the cross-validation errors for
# difference in the CV errors between the two methods.
t.test(err.cv[,1], err.cv[,2], paired=TRUE ) # no difference MLR and NN
t.test(err.cv[,1], err.cv[,3], paired=TRUE) # mlr and pcr are statistically different
t.test(err.cv[,2], err.cv[,3], paired=TRUE)      #no difference nn and pcr

anova <- aov(values ~ ind , data = err.cv.df)
summary(anova )
TukeyHSD(anova)
```

```
        Paired t-test

data:  err.cv[, 1] and err.cv[, 2]
t = -0.46626, df = 9, p-value = 0.6521
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.02974027  0.01957571
sample estimates:
mean of the differences
          -0.005082281


        Paired t-test

data:  err.cv[, 1] and err.cv[, 3]
t = -4.4651, df = 9, p-value = 0.001566
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.029584044 -0.009687724
sample estimates:
mean of the differences
           -0.01963588


        Paired t-test

data:  err.cv[, 2] and err.cv[, 3]
t = -1.3372, df = 9, p-value = 0.214
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.03917433  0.01006712
sample estimates:
mean of the differences
            -0.0145536
```



T-test result:

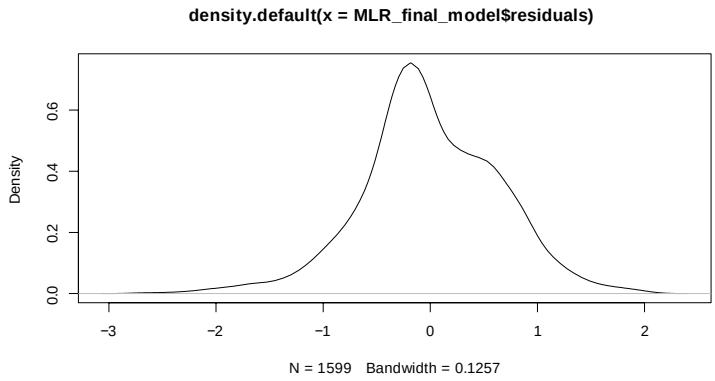
looking at the t-test result, we see that multiple linear regression and neural network are not statistically significant. also no difference between neural network and principal component regression since the p-value is big.However, there is a difference between multiple linear regression and principal component regression, because the p-value is less than 0.05. Therefore, we choose MLR as our final model in order to predict the rate of the wine because it has the smallest mean square error.

Fit MLR in the the whole data:

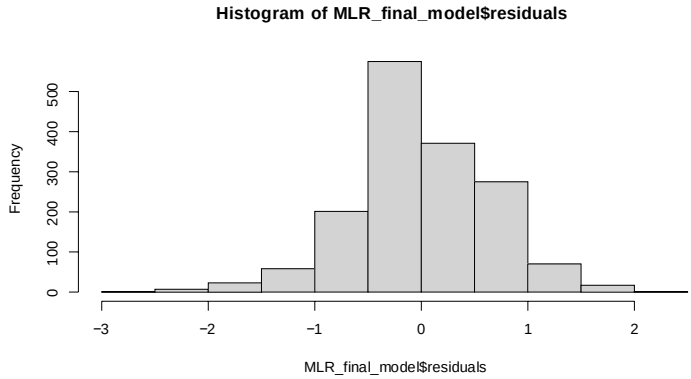
```
# (g) Fit the model to the whole data to get your final model for Step 3 of the objective.
# Show summary of your model fit.

MLR_final_model = lm(quality~., data = wine)
summary(MLR_final_model)
```

```
##
## Call:
## lm(formula = quality ~ ., data = wine)
##
## 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
```

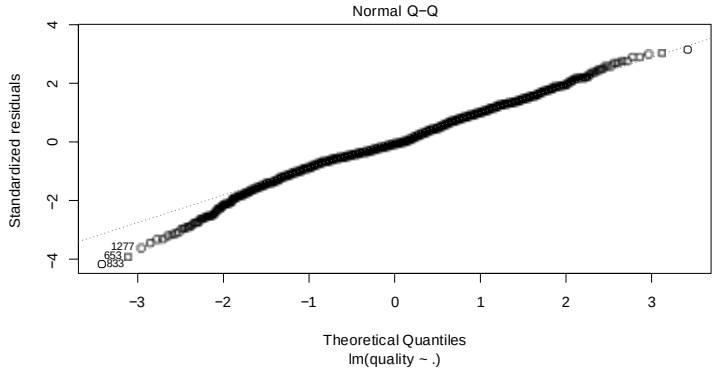
```
plot(density(MLR_final_model$residuals))
```




```
hist(MLR_final_model$residuals)
```




```
plot(MLR_final_model , which = 2)
```



from the MLR summary we can see that some variable are not significant in the our model.Hence we are going to take them out. for this data the following variable are not significant: -fixed acidity. -citric acid. -residual sugar. -density. Moreover, the QQ-plot shows that the data is normally distributed,the density plot(bell shape) and the histogram confirm the normality of the data.

Final Model:

our final model will be as following :

quality = volatile acidity + chlorides + free sulfur dioxide + total sulfur dioxide + pH + sulphates + alcohol

```
MLR_final_model = lm(quality~ `volatile acidity` + chlorides + `free sulfur dioxide`     + `total sulfur d
summary(MLR_final_model)
```

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