Performance comparison of models trained with Decision Trees and Multivariate Regression methods

Machine Learning and Applications

Tufan Bostan

17 04 2021

In this work, multinomial regression model performance and decision trees performance will be compared using R. Data to be used here is "wine" from “rattle.data” package. The wine dataset contains the results of a chemical analysis of wines grown in a specific area of Italy. Three types of wine are represented in the 178 samples, with the results of 13 chemical analyses recorded for each sample. The Type variable has been transformed into a categoric variable. The data contains no missing values and consist of only numeric data, with a three-class target variable (Type) for classification. First, some descriptive statistics will be obtained and interpreted. Then, the data will be split into two parts which are “Train” and “Test”. Then using the “Train” part of data models will be generated using both methods. Finally, performance of the models on train and test sets will be interpreted and Decision Trees and Multivariate Regression methods performances will be compared.

Data Structure

First of all, “rattle.data” package installed to access “wine” data (`install.packages("rattle.data")`). Now “wine” data introduced to R. The “wine” data assigned to “data”. All variables were correctly classified. First 6 observations of the data are listed below. Then, structure of the data displayed. (`str(data)`) After that, data summary displayed to gather more information about variables. (`summary(data)`)

Also, some packages that can be used in this study have been installed.

```
#install.packages("e1071")
#install.packages("rattle.data")
#install.packages("rpart")
#install.packages("rpart.plot")
#install.packages("nnet")
#install.packages("caret")
#Obtaining Data:
library(rattle.data)

data=rattle.data::wine          #The data assigned to "data".
head(data)

##   Type Alcohol Malic  Ash Alcalinity Magnesium Phenols Flavanoids Nonflavanoids
## 1    1   14.23  1.71 2.43       15.6       127    2.80       3.06          0.28
## 2    1   13.20  1.78 2.14       11.2       100    2.65       2.76          0.26
## 3    1   13.16  2.36 2.67       18.6       101    2.80       3.24          0.30
## 4    1   14.37  1.95 2.50       16.8       113    3.85       3.49          0.24
## 5    1   13.24  2.59 2.87       21.0       118    2.80       2.69          0.39
## 6    1   14.20  1.76 2.45       15.2       112    3.27       3.39          0.34
##   Proanthocyanins Color  Hue Dilution Proline
## 1            2.29  5.64 1.04     3.92    1065
## 2            1.28  4.38 1.05     3.40    1050
## 3            2.81  5.68 1.03     3.17    1185
## 4            2.18  7.80 0.86     3.45    1480
## 5           1.82  4.32 1.04     2.93     735
## 6           1.97  6.75 1.05     2.85    1450

#Structure of The Data:
str(data)

## 'data.frame':    178 obs. of  14 variables:
##  $ Type           : Factor w/ 3 levels "1","2","3": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Alcohol        : num  14.2 13.2 13.2 14.4 13.2 ...
##  $ Malic          : num  1.71 1.78 2.36 1.95 2.59 1.76 1.87 2.15 1.64 1.35 ...
##  $ Ash            : num  2.43 2.14 2.67 2.5 2.87 2.45 2.45 2.61 2.17 2.27 ...
##  $ Alcalinity     : num  15.6 11.2 18.6 16.8 21 15.2 14.6 17.6 14 16 ...
##  $ Magnesium      : int  127 100 101 113 118 112 96 121 97 98 ...
##  $ Phenols        : num  2.8 2.65 2.8 3.85 2.8 3.27 2.5 2.6 2.8 2.98 ...
##  $ Flavanoids     : num  3.06 2.76 3.24 3.49 2.69 3.39 2.52 2.51 2.98 3.15 ...
##  $ Nonflavanoids  : num  0.28 0.26 0.3 0.24 0.39 0.34 0.3 0.31 0.29 0.22 ...
##  $ Proanthocyanins: num  2.29 1.28 2.81 2.18 1.82 1.97 1.98 1.25 1.98 1.85 ...
##  $ Color          : num  5.64 4.38 5.68 7.8 4.32 6.75 5.25 5.05 5.2 7.22 ...
##  $ Hue            : num  1.04 1.05 1.03 0.86 1.04 1.05 1.02 1.06 1.08 1.01 ...
##  $ Dilution       : num  3.92 3.4 3.17 3.45 2.93 2.85 3.58 3.58 2.85 3.55 ...
##  $ Proline        : int  1065 1050 1185 1480 735 1450 1290 1295 1045 1045 ...

summary(data)

##  Type      Alcohol          Malic            Ash          Alcalinity
##  1:59   Min.   :11.03   Min.   :0.740   Min.   :1.360   Min.   :10.60
##  2:71   1st Qu.:12.36   1st Qu.:1.603   1st Qu.:2.210   1st Qu.:17.20
##  3:48   Median :13.05   Median :1.865   Median :2.360   Median :19.50
##         Mean   :13.00   Mean   :2.336   Mean   :2.367   Mean   :19.49
##         3rd Qu.:13.68   3rd Qu.:3.083   3rd Qu.:2.558   3rd Qu.:21.50
##         Max.   :14.83   Max.   :5.800   Max.   :3.230   Max.   :30.00
##    Magnesium         Phenols        Flavanoids    Nonflavanoids
##  Min.   : 70.00   Min.   :0.980   Min.   :0.340   Min.   :0.1300
##  1st Qu.: 88.00   1st Qu.:1.742   1st Qu.:1.205   1st Qu.:0.2700
##  Median : 98.00   Median :2.355   Median :2.135   Median :0.3400
##  Mean   : 99.74   Mean   :2.295   Mean   :2.029   Mean   :0.3619
##  3rd Qu.:107.00   3rd Qu.:2.800   3rd Qu.:2.875   3rd Qu.:0.4375
##  Max.   :162.00   Max.   :3.880   Max.   :5.080   Max.   :0.6600
##  Proanthocyanins     Color             Hue            Dilution
##  Min.   :0.410   Min.   : 1.280   Min.   :0.4800   Min.   :1.270
##  1st Qu.:1.250   1st Qu.: 3.220   1st Qu.:0.7825   1st Qu.:1.938
##  Median :1.555   Median : 4.690   Median :0.9650   Median :2.780
##  Mean   :1.591   Mean   : 5.058   Mean   :0.9574   Mean   :2.612
##  3rd Qu.:1.950   3rd Qu.: 6.200   3rd Qu.:1.1200   3rd Qu.:3.170
##  Max.   :3.580   Max.   :13.000   Max.   :1.7100   Max.   :4.000
##     Proline
##  Min.   : 278.0
##  1st Qu.: 500.5
##  Median : 673.5
##  Mean   : 746.9
##  3rd Qu.: 985.0
##  Max.   :1680.0
```

The data has 14 variables and 178 observations. There is a factor and 13 numeric variables. Here, the target variable will be "Type". The other variables are the independent variables, to be used to predict the "Type".

Type The type of wine, into one of three classes, 1 (59 obs), 2(71 obs), and 3 (48 obs).

- Alcohol: Alcohol
- Malic: Malic acid
- Ash: Ash
- Alcalinity: Alcalinity of ash

- Magnesium: Magnesium
- Phenols: Total phenols
- Flavanoids: Flavanoids
- Nonflavanoids: Nonflavanoid phenols
- Proanthocyanins: Proanthocyanins
- Color: Color intensity.
- Hue: Hue
- Dilution: D280/OD315 of diluted wines.
- Proline: Proline

Here, the information about the variables in the data is listed as found in its source. Descriptive statistics of 13 numerical variables are given. In addition, the levels of the factor variable, which is our target variable, are also given. For example, Alcohol has 13.05 mean and ranges 11.03 to 14.83. "Type" is representing target variable. it's a factor, has 3 levels which are "1", "2" and "3". They were observed 59, 71 and 48 times, respectively. Here, it is possible to say that the number of observations at the "1" and "3" levels of the "Type" variable, are close to each other. However, since "2" has more observations than the other levels, the imbalance problem may occur. Also some features can have outliers it may cause low accuracy.

Splitting Data

A sample was created using "wine" data. This sample has two parts one of them is going to be used for training to model and the other part for testing to the model. These are assigned to variables named `train` and `test`, respectively. Also, using `table(train$status);table(test$status)`, how many observations they contain is shown.

```
#Splitting Data
set.seed(2380)                               #to get same sample every time
index <- sample(nrow(data),nrow(data)*0.8)   #Splitting Data
train <- data[index,]                        #assigning observations for train
test <- data[-index,]                        #assigning observations for test
table(train$Type);table(test$Type)           #displaying variable sizes

##
##  1  2  3
## 51 56 35

##
##  1  2  3
##  8 15 13
```

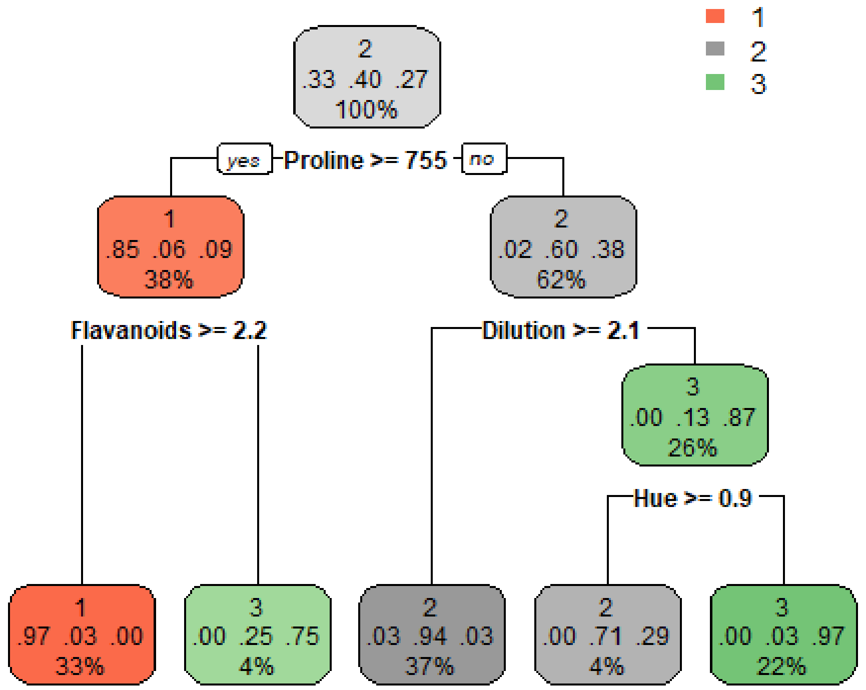
`train` has 51 "1", 56 "2" and 35 "3" observations and `test` 8 "1", 15 "2" and 13 "3" observations. After this process, the MLRM and DTM is ready to be generated.

Decision Trees Method

Here the decision there were displayed. 2 was the most observed in the test part of the data so in root node, 2 is observed. The colors of the nodes are depending on the levels and the levels' colors shown on top right. Here the nodes take the color of the most observed level. For example, in the first node (root node), %40 of them is 2 so the node has grey color.

```
library(rpart)
library(rpart.plot)
```

```
model <- rpart(Type ~ ., data = data, method = "class")
rpart.plot(model)
```



The ratios in the boxes indicate what level from which level. As mentioned before %40 of them is 2, %33 of them is 1 and %27 of them is 3. Percentages are indicated on the last line of each box. These percentages indicate what percentage of all data is in that node. For root node, this percentage is 100 because the data did not splitted yet. If the Proline is more or equal than 755 then it goes left. If its less than 755 then it goes right. Let’s consider Proline is more than 755. Then it goes to left. That box colored as red which means, 1 is most observed level. And the percentage that written of last row of this box %38 so the main data splitted into two parts and %38 of them on the left and %62 of them on the right. If we consider Flavonoids is higher than 2.2 then it goes left again. This node (leaf node) also red which means 1 is most observed level. As a result, %33 of the data has more than 755 Proline and more than 2.2 Flavonoids, and the model predicts them as 1 whit 0.97 ratio.

```
#Trainig DT
model_dt <- rpart(Type ~., method = "class", data = train)
#Pedtormance of the Model on Train and Test Set:
#FOR TRAIN
pred_labels_dt_train <- predict(model_dt, train, type = "class")
conf_mat_dt_train <- table(pred_labels_dt_train, train$Type)
print(c("Train:",sum(diag(conf_mat_dt_train))/sum(conf_mat_dt_train)))

## [1] "Train:"            "0.936619718309859"

#FOR TEST
pred_labels_dt_test <- predict(model_dt, test, type = "class")
conf_mat_dt_test <- table(pred_labels_dt_test, test$Type)
print(c("Test:",sum(diag(conf_mat_dt_test))/sum(conf_mat_dt_test)))

## [1] "Test:"             "0.861111111111111"
```

Now, the performance of the model calculated. The model classifies the “train” approximately %93 correctly. Same process made on “testing” data. The model classifies the “test” approximately

%86 correctly. The accuracy ratio that we obtained using the “train” is larger than “test”. We might suspect of overfitting problem. Training whit more data, removing some features or training an ensemble model can be applied to remove overfitting problem.

Multi Linear Regression Model

```
train$Type <- relevel(train$Type, ref = "1")            #Reference level were determined.
library(nnet)                                           #To use "multinom()" function.
model_mlrm <- multinom(Type ~., data = train, trace = F)  #Model generated by train data
predicted_probs <- predict(model_mlrm,type = "probs")

#Pedtormance of the Model on Train and Test Set:
#For TRAIN
predicted_probs_train <- predict(model_mlrm, type = "probs")
predicted_class_train <- colnames(predicted_probs_train)[apply(predicted_probs_train, 1,whi
ch.max)]
conf_mat_mlr_train <- table(predicted_class_train, train$Type)
print(c("Train:",sum(diag(conf_mat_mlr_train))/sum(conf_mat_mlr_train)))

## [1] "Train:" "1"

#For TEST
predicted_probs_test <- predict(model_mlrm, test, type = "probs")
predicted_class_test <- colnames(predicted_probs_test)[apply(predicted_probs_test, 1, which
.max)]
conf_mat_mlr_test <- table(predicted_class_test, test$Type)
print(c("Train:",sum(diag(conf_mat_mlr_test))/sum(conf_mat_mlr_test)))

## [1] "Train:"            "0.944444444444444"
```

Here, the performance of the MLRM were obtained. The model classifies the “train” %100 correctly and “test” approximately %94 correctly. The accuracy ratio that we obtained using the “train” is larger than “test”. In this model we might also suspect of overfitting problem. Training whit more data, removing some features or training an ensemble model can be applied to remove overfitting problem.

Decision Trees vs Multi Linear Regression

Finally, both models were generated and their accuracy were calculated. Now, these models will be compared by their accuracy, sensitivity and specificity. These ratios were obtained below.

```
library(caret)

#-----------Decision Trees---------------
confusionMatrix(
factor(pred_labels_dt_test, levels = 1:3),
factor(test$Type, levels = 1:3)
)

## Confusion Matrix and Statistics
##
##           Reference
## Prediction  1  2  3
##          1  8  1  0
##          2  0 13  3
##          3  0  1 10
##
## Overall Statistics
##
##                Accuracy : 0.8611
##                  95% CI : (0.705, 0.9533)
##     No Information Rate : 0.4167
##     P-Value [Acc > NIR] : 4.67e-08
##
##                   Kappa : 0.786
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3
## Sensitivity            1.0000   0.8667   0.7692
## Specificity            0.9643   0.8571   0.9565
## Pos Pred Value         0.8889   0.8125   0.9091
## Neg Pred Value         1.0000   0.9000   0.8800
## Prevalence             0.2222   0.4167   0.3611
## Detection Rate         0.2222   0.3611   0.2778
## Detection Prevalence   0.2500   0.4444   0.3056
## Balanced Accuracy      0.9821   0.8619   0.8629
```

```
#---------------------MLR------------------------
confusionMatrix(
factor(predicted_class_test, levels = 1:3),
factor(test$Type, levels = 1:3)
)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  1  2  3
##          1  8  1  0
##          2  0 13  0
##          3  0  1 13
##
## Overall Statistics
##
##                Accuracy : 0.9444
##                  95% CI : (0.8134, 0.9932)
##     No Information Rate : 0.4167
##     P-Value [Acc > NIR] : 2.641e-11
##
##                   Kappa : 0.915
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3
## Sensitivity            1.0000   0.8667   1.0000
## Specificity            0.9643   1.0000   0.9565
## Pos Pred Value         0.8889   1.0000   0.9286
## Neg Pred Value         1.0000   0.9130   1.0000
## Prevalence             0.2222   0.4167   0.3611
## Detection Rate         0.2222   0.3611   0.3611
## Detection Prevalence   0.2500   0.3611   0.3889
## Balanced Accuracy      0.9821   0.9333   0.9783
```

According to their overall statistics DT model has 0.8611 accuracy and MLR has 0.9444 accuracy which is higher than DT. Thus, we can clearly say that the MLR method were more successful on predicting wine Types. Also, both models’ sensitivities and specificities are pretty high so both of them are good models but since MLR has better scores then MLR is better for this work.