WARWICK BUSINESS SCHOOL
THE UNIVERSITY OF WARWICK

For the Change Makers

Dr. Iman Ahmadi
Assistant Prof. of Marketing

Marketing & Strategy Analytics:

Supervised Learning: Regression Trees

Overview of Regression Tree

Splitting criteria

- At any given point in tree, choose to split on
  - independent variable
  - value within the respective independent variable

that maximizes reduction in Sum of Squared Errors (SSE)

Predicted value

- **<u>Average</u>** value of observations (of dependent variable) in each leaf node

Lantz, B. (2015) *Machine Learning with R (Second edition)*. Birmingham: Packt Publishing. Chapter 6.

Using Reduction in Sum of Squared Errors (SSE) to Determine Appropriate Split

- **Basic idea of SSE similar to basic idea of Information Gain**
- **Reduction in SSE for a possible variable (*F*) is difference between SSE in segment**
  - before the split
  - after the split: weighted SSE

$$SSE\ before\ split = \sum_{j=1}^{n} \left(y_j - \hat{y}_j\right)^2$$

$$SSE\ after\ split = \sum_{i=1}^{c} \sum_{j=1}^{n_i} \left(y_{ji} - \hat{y}_{ji}\right)^2$$

j: index for observation
i: index for subset (after split)
n: number of observations;
n_i: number of observations in subset i;
c: number of subsets after split;
$n_1 + \cdots + n_c = n$;
y_j: the dependent variable before split;
$\hat{y}_j$: the predicted value before split;
y_{ji}: the dependent variable (in subset i) after split;
$\hat{y}_{ji}$: the predicted value (in subset i) after split

$$Reduction\ in\ SSE(F) = SSE\ before\ split\text{-}\ SSE\ after\ split$$

- $\boldsymbol{\hat{y}_{ji}}$, **i.e., predicted value for observation j in subset i**
  - For example, in regression tree: $\boldsymbol{\hat{y}_{ji}} = \frac{\mathbf{1}}{\boldsymbol{n_i}} \sum_{j=1}^{n_i} y_{ji}$

Miguel Forte, R. (2015) *Mastering Predictive Analytics with R*. Birmingham: Packt Publishing. Chapter 6.

Using Standard Deviation Reduction (SDR) to Determine Appropriate Split

- **SDR for a possible (independent) variable (*F*) is difference between standard deviation (SD) in segment**
  - before the split
  - after the split: weighted sum of SD

$$SD\ before\ split = sd(S)$$

$$SD\ after\ split = \sum_{i=1}^{c} \frac{|S_i|}{|S|} sd(S_i)$$

$$SDR(F) = sd(S) - \sum_{i=1}^{c} \frac{|S_i|}{|S|} sd(S_i)$$

S: subset of data;
c: number of subsets after split;
$|S_i|$: size of subset i generated from S;
$|S|$: size of subset S;
$sd(S)$: standard deviation of subset S;
$sd(S_i)$: standard deviation of subset i generated from S

Miguel Forte, R. (2015) *Mastering Predictive Analytics with R*. Birmingham: Packt Publishing. Chapter 6.

Exercise 6.2 – Predicting Quality of Wines (I/II)

- A wine making is a profitable but challenging and competitive business. Wine industry has heavily invested in ways to assist wine makers.
- You are asked to come up with a model that helps to predict the quality of wines and identify key factors that affect quality of a wine.
- Use dataset "Wine" and:
  - build a regression tree in R using rpart command (dependent variable: 'quality')
  - interpret your results and plot them

("Wine" R code)

Exercise 6.2 – Predicting Quality of Wines (II/II)

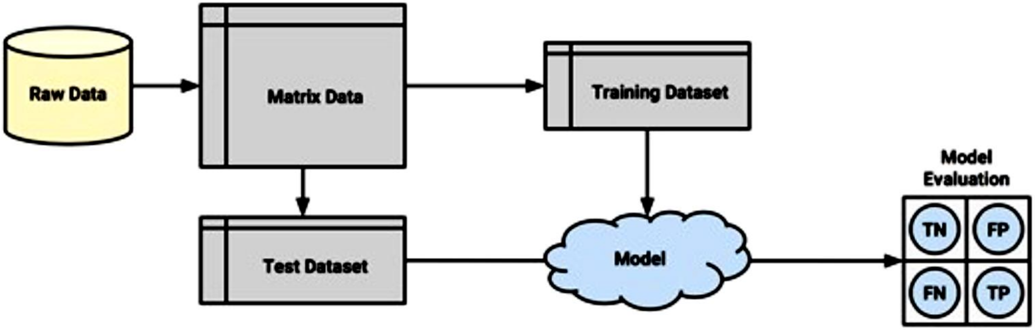
- **Dataset includes:**
  - 4,898 observations
  - 12 variables
    - Independent variables: 11 chemical properties of samples:
      - laboratory analysis of fixed acidity, volatile acidity, and citric acid
      - sugar content
      - chlorides
      - free sulfur dioxide and total sulfur dioxide
      - density, alcohol, pH, and sulphates
    - Dependent variable: quality scale ranging from 0 (very bad) to 10 (excellent)

("Wine" R code)

Holdout Method

Holdout Method

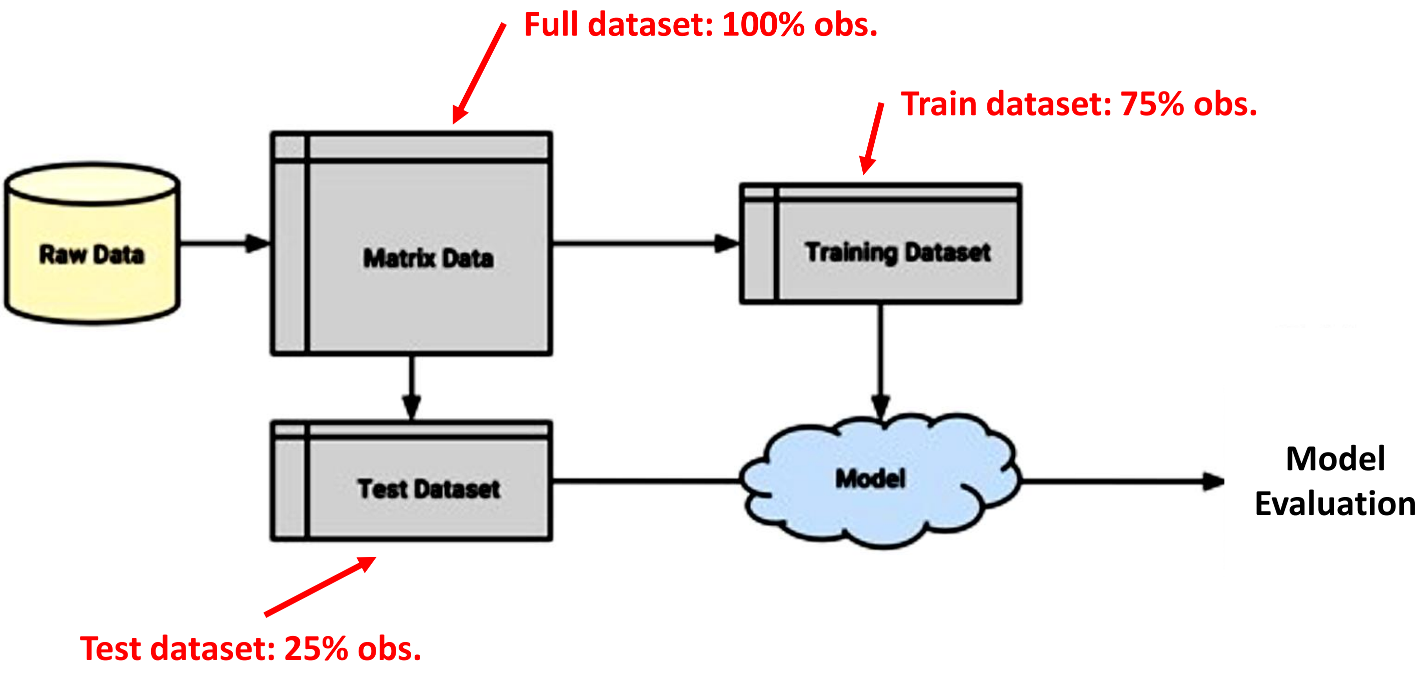
- Holdout method is procedure of splitting data into training and test subsets
  - model builds upon training dataset
  - model then predicts upon test dataset
  - keep one-third to 10% of the whole data (as rule of thumb) for testing



Lantz, B. (2015) *Machine Learning with R (Second edition)*. Birmingham: Packt Publishing. Chapter 10.

Application of Holdout Method in: Regression Trees

Visualizing Holdout Method: "Predicting Quality of Wines"



Application of Holdout Method on "Quality of Wines"

- Data:
  - 4,898 observations
  - 12 variables
    - 11 chemical properties of samples:
      - laboratory analysis of fixed acidity, volatile acidity, and citric acid
      - sugar content
      - chlorides
      - free sulfur dioxide and total sulfur dioxide
      - density, alcohol, pH, and sulphates
    - quality scale ranging from zero (very bad) to 10 (excellent)

- Split data
  - 75% for training (i.e., observations 1 to 3,750)
  - 25% for testing (i.e., observations 3,751 to 4,898)

- Evaluate the model

("Wine_Holdout" R code)

Reminder: "Quality of Wines"

Step 1: explore your data

```
str(wine)
```

```
'data.frame':      4898 obs. of  12 variables:
 $ fixed.acidity       : num  6.7 5.7 5.9 5.3 6.4 7 7.9 6.6 7 6.5 ...
 $ volatile.acidity    : num  0.62 0.22 0.19 0.47 0.29 0.14 0.12 0.38...
 $ citric.acid         : num  0.24 0.2 0.26 0.1 0.21 0.41 0.49 0.28...
 $ residual.sugar      : num  1.1 16 7.4 1.3 9.65 0.9 5.2 2.8 2.6 3.9 ...
 $ chlorides           : num  0.039 0.044 0.034 0.036 0.041 0.037...
 $ free.sulfur.dioxide : num  6 41 33 11 36 22 33 17 34 40 ...
 $ total.sulfur.dioxide: num  62 113 123 74 119 95 152 67 90 130 ...
 $ density             : num  0.993 0.999 0.995 0.991 0.993 ...
 $ pH                  : num  3.41 3.22 3.49 3.48 2.99 3.25 3.18 3.21...
 $ sulphates           : num  0.32 0.46 0.42 0.54 0.34 0.43 0.47 0.47...
 $ alcohol             : num  10.4 8.9 10.1 11.2 10.9 ...
 $ quality             : int  5 6 6 4 6 6 6 6 6 7 ...
```

Step 2: Create Train and Test Dataset

Let's take the first 75% of observations as train and the rest as test dataset

```
wine_train <- wine[1:3750, ]
wine_test <- wine[3751:4898, ]
```

Step 3: Training the Model on the Train Dataset

```
library(rpart)
m.rpart <- rpart(quality ~ ., data = wine_train)
```

Note: As usual, you can get simple/detailed information about the tree you have made using:

```
m.rpart
summary(m.rpart)
```

Step 4: Apply your Model on Test Dataset and Predict

Predict the quality of wine in the test dataset (i.e., remaining 25% obs.):

```
p.rpart <- predict(m.rpart, wine_test)
```

Step 5: Evaluating Model Performance Using Test Dataset (I/II)

Compare summary for actual quality with predicted quality in the remaining 25% obs.:

```
summary(wine_test$quality)
```

```
 Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 3.000   5.000   6.000   5.901   6.000   9.000
```

```
summary(p.rpart)
```

```
 Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 4.545   5.563   5.971   5.893   6.202   6.597
```

Step 5: Evaluating Model Performance Using Test Dataset (II/II)

Summarize your perditions using MAE:

$$MAE(Mean\ Absolute\ Error) = \frac{1}{n}\sum_{i=1}^{n}|e_i|,$$ e_i: error for prediction of wine *i*

Function for MAE and check your model performance on the test dataset:

```
MAE <- function(actual, predicted) {
  mean(abs(actual - predicted))
}

MAE(wine_test$quality, p.rpart)
```

```
[1] 0.5872652
```

On average, you are making 0.59 error in your predictions for the quality of wine (in your test dataset)

Thank You!

Iman.Ahmadi@wbs.ac.uk

Room No.: 3.207