PSS718 - Data Mining

Lecture 5 - Transforming Data

Asst.Prof.Dr. Burkay Genç

Hacettepe University

October 23, 2016

Improving the performance of a model

- To improve the performance of a model, we mostly improve the data
  - Source additional data
  - Clean up the data
  - Deal with missing values
  - Transform the data
  - Analyze the data to choose better variables

The ACM KDD Cup

Building models from the right data is crucial to the success of a data mining project.

The ACM KDD Cup, an annual Data Mining and Knowledge Discovery competition, is often won by a team that has placed a lot of effort in preprocessing the data supplied.

ACM KDD 2009

- Orange supplied data related to customer relationship management
- 50,000 observations with much missing data
- Each observation recorded values for 15,000 (anonymous) variables
- Three target variables were to be modeled
- You really need to pre-process this before mining!

Data cleaning

- When collecting data, it is not possible to ensure that it is perfect
- There are many reasons for the data to be dirty:
  - Simple data entry errors
  - Decimal points can be incorrectly placed
  - There can be inherent error in any counting or measuring device
  - External factors that cause errors to change over time

A number of simple steps

- Most cleaning will be done during exploration
- Check frequency counts and histograms for anomalies
- Check very low category counts for categoric variables
- Check names and adresses, these usually have many versions

Example

Genc vs Genç
Çırağan vs Cırağan vs Ciragan vs ...
Hacettepe Üniversitesi vs Hacettepe University vs University of Hacettepe

Missing data

- Missing data is a common feature of any dataset
- Sometimes there is no information available to populate some value
- Sometimes the data has simply been lost
- Sometimes the data is purposefully missing because it does not apply to a particular observation
- For whatever reason the data is missing, we need to understand and possibly deal with it

Some examples

- Use of sentinels for missing data
  - Symbolic values: 999, 1 Jan 1900, Earth (for an address)
  - Negative values where a positive is necessary: -1
  - Special characters: *, #, $, %, -
- Simply missing data:
- Character replacements: None, Missing, Null, Absent

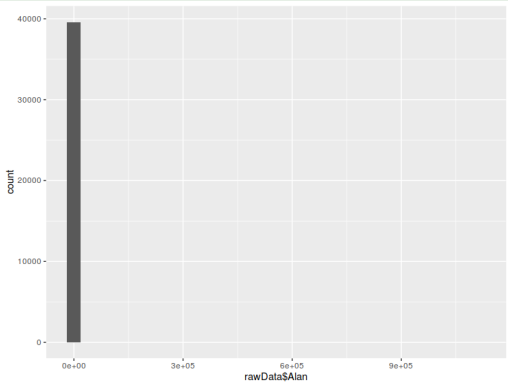
Outliers

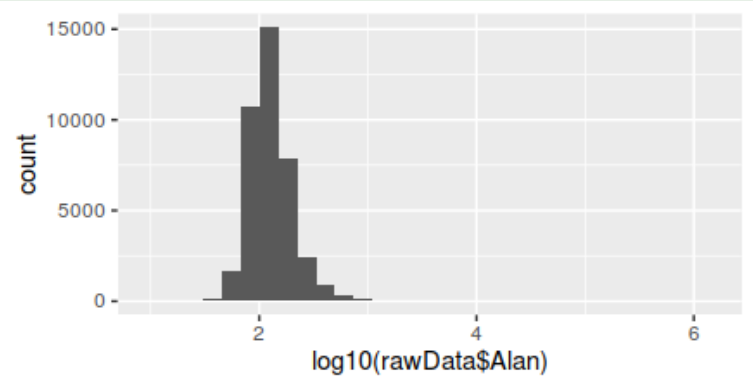
Definition

An outlier is an observation that has values for the variables that are quite different from most other observations.

Example

```
> summary(rawData$Alan)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
     10      95     120     591     160 1111000
```





Outlier vs high variance

Hawkins (1980)

An outlier is an observation that deviates so much from other observations as to arouse suspicion that it was **generated by a different mechanism**.

- Extreme weather conditions
- Extremely rich people
- Extremely short people (midgets)
- We have to be careful in deciding what is an outlier and what is not

Looking for outliers

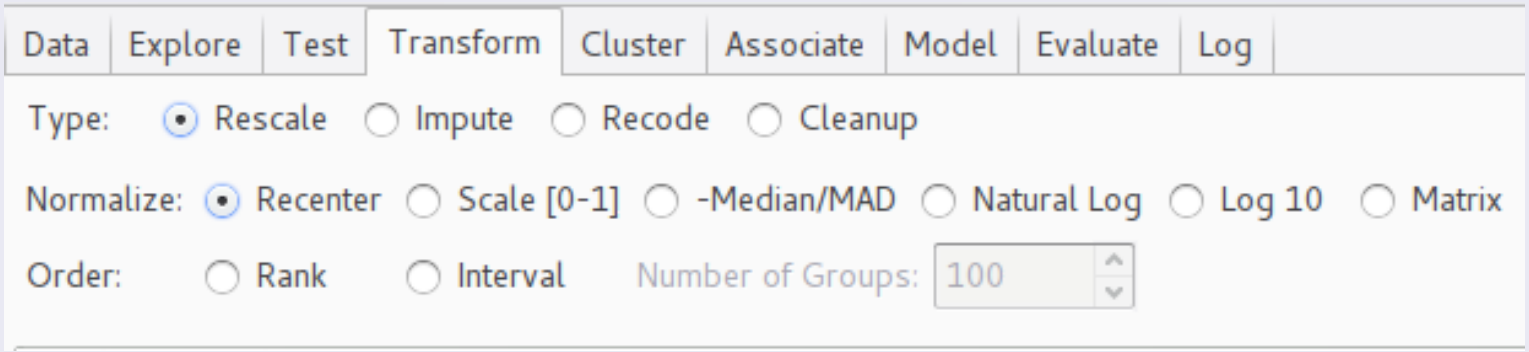
- Sometimes outliers are what we are looking for:
  - Fraud in income tax
  - Fraud in insurance
  - Fraud in medical payment and medication expenses
  - Marketing fraud

Variable selection

- By removing irrelevant variables from the modeling process, the resulting models can be made more robust
- Some variables will also be found to be quite related to other variables
- Various techniques exist:
  - Random subset selection
  - Principal component analysis
  - Variable importance measures of random forests
  - ...

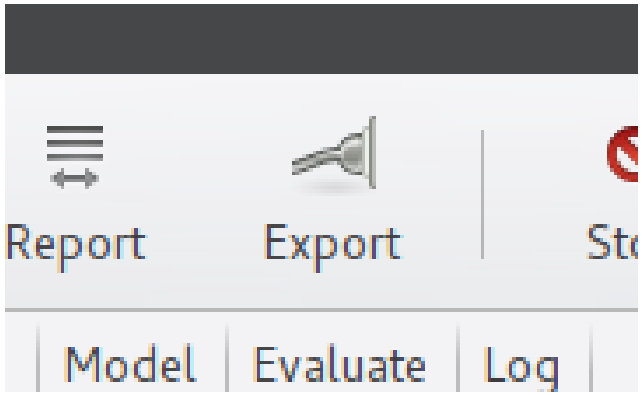
Rattle and transformation

Rattle supports many techniques for transforming data.



Don't forget

- Load Data
- Explore
- Transform
- **Save!**



Alternative way of saving

Log saving

Save your log to a script!
This will allow you to rerun the script to generate the modified dataset.
Or apply the modification to another dataset!
You can also modify the way of generation!

Rescaling

- Different model builders will have different assumptions on the data from which the models are built
- When building a cluster, ensure all variables have the same scale

Example

Observation 1: Income -> $10,000 and Age -> 30
Observation 2: Income -> $10,500 and Age -> 70
Observation 3: Income -> $9,000 and Age -> 32
Which two observations are closest?

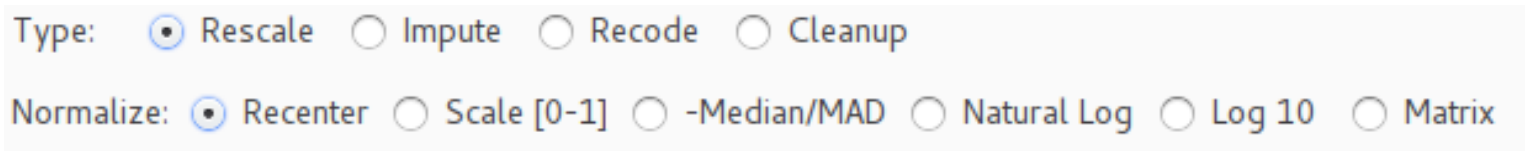
Normalization

```
> df
  income age
1  10000  30
2  10500  70
3   9000  32
> dist(df)
          1         2
2  501.5974
3 1000.0020 1500.4813
```

```
> df$income <- (df$income - mean(df$income)) / sd(df$income)
> df$age <- (df$age - mean(df$age)) / sd(df$age)
> df
      income        age
1  0.2182179 -0.6211496
2  0.8728716  1.1535635
3 -1.0910895 -0.5324139
```

```
> dist(df)
         1        2
2 1.891607
3 1.312311 2.588371
```

Normalization



- Recenter: uses a so-called Z score, which subtracts the mean and divides by the standard deviation
- Scale [0-1]: rescaling our data to be in the range from 0 to 1
- Median/MAD: a robust rescaling around zero using the median
- Log 10: Obvious
- Matrix: transforming multiple variables with one divisor
- Rank: Order and rank the observations
- Interval: Group the observations into a predefined amount of bins

Recenter

Definition (Recenter)

This is a common normalisation that re-centres and rescales our data. The usual approach is to subtract the mean value of a variable from each observation's value of the variable (to recentre the variable) and then divide the values by their standard deviation (calculating the square root of the sum of squares), which rescales the variable back to a range within a few integer values around zero.

Example

```
> df$RRC_Temp3pm <- scale(df$Temp3pm)
```

Scale [0-1]

Definition (Scale [0-1])

This is done by subtracting the minimum value from the variable's value for each observation and then dividing by the difference between the minimum and the maximum values.

Example

```
> library(reshape)
> df$R01_Temp3pm <- rescaler(df$Temp3pm, "range")
```

Median/MAD

Definition (Median/MAD)

This option for re-centering and rescaling our data is regarded as a robust (to outliers) version of the standard Recenter option. Instead of using the mean and standard deviation, we subtract the median and divide by the so-called median absolute deviation (MAD).

Example

```
> library(reshape)
> df$RMD_Temp3pm <- rescaler(df$Temp3pm, "robust")
```

Natural Logarithm

- Used when the distribution of a variable is quite skewed
- Maps a very broad range into a narrower range
- Outliers are easily handled
- Default is to log in base e (natural logarithm)
- Be careful for $\log 0 = -\infty$ and log of negative values

Example

```
> df$RLG_Temp3pm <- log(df$Temp3pm)
> df$RLG_Temp3pm[df$RLG_Temp3pm == -Inf] <- NA
```

Rank

Definition (Rank)

The Rank will convert each observation's numeric value for the identified variable into a ranking in relation to all other observations in the dataset. A rank is simply a list of integers, starting from 1, that is mapped from the minimum value of the variable, progressing by integer until we reach the maximum value of the variable.

Example

```
> library(reshape)
> df$RRK_Temp3pm <- rescaler(df$Temp3pm, "rank")
```

Interval

Definition (Interval)

An Interval transform recodes the values of a variable into a rank order between 0 and 100.

What is Imputation?

Definition (Imputation)

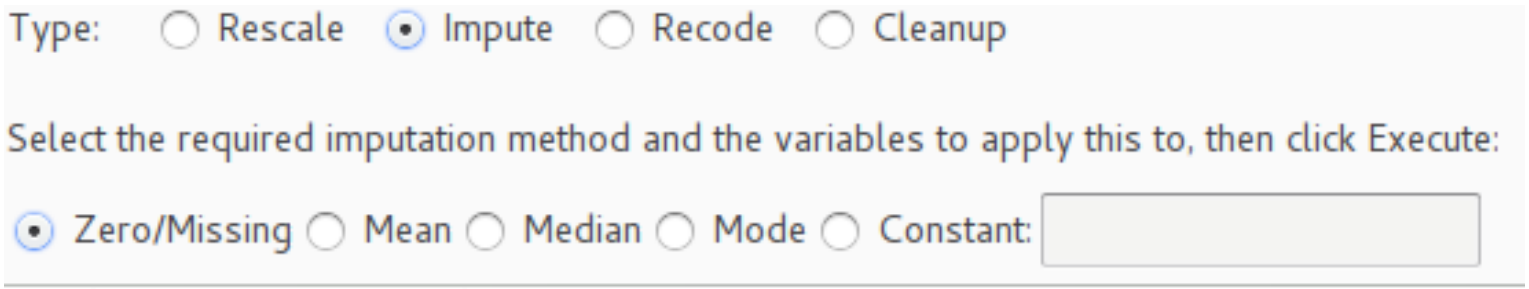
Imputation is the process of filling in the gaps (or missing values) in data.

- Data is missing for many different reasons, and it is important to understand why
- Imputation can be questionable because, after all, we are inventing data

How to do it?

- Replace with a particular value (helps with regression)
- Add an additional variable to denote when values are missing (helps with models identifying the importance of the missing values)
- Some models are not troubled with missing values, e.g., decision trees

How Rattle does it?



Zero/Missing

Definition (Zero/Missing)

The simplest imputations involve replacing all missing values for a variable with a single value. This makes the most sense when we know that the missing values actually indicate that the value is 0 rather than unknown.

Example

```
> df$IZR_Sunshine <- df$Sunshine
> df$IZR_Sunshine[is.na(df$IZR_Sunshine)] <- 0
```

Mean/Median/Mode

Definition (Mean/Median/Mode)

Often a simple, if not always satisfactory, choice for missing values that are known not to be zero is to use some "central" value of the variable. This is often the mean, median, or mode, and thus usually has limited impact on the distribution.

- no skewness -> use mean
- skewed -> use median
- categoric variables -> use mode (the most frequent category)

Example

```
> df$IMN_Sunshine <- df$Sunshine
> df$IMN_Sunshine[is.na(df$IMN_Sunshine)] <-
mean(df$Sunshine, na.rm=TRUE)
```

Constant

Definition (Constant)

This choice allows us to provide our own default value to fill in the gaps. This might be an integer or real number for numeric variables, or else a special marker or the choice of something other than the majority category for categoric variables.

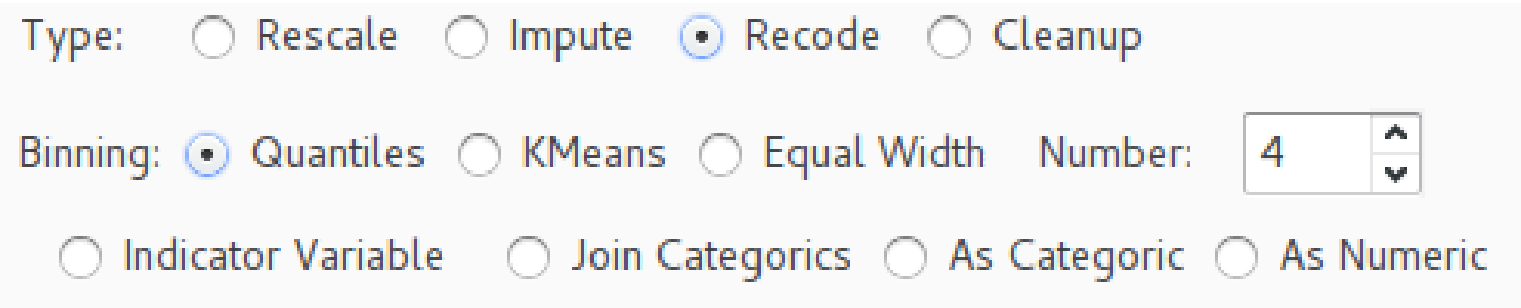
Example

```
> df$ICN_Sunshine <- df$Sunshine
> df$ICN_Sunshine[is.na(df$IZR_Sunshine)] <- 0
```

What is Recoding

Definition (Recoding)

Recoding provides numerous remapping operations, including binning and transformations of the type of the data.



Binning

Definition (Binning)

Binning is the operation of transforming a continuous numeric variable into a specific set of categoric values based on the numeric values.

- Age into AgeGroups
- Temperature into Low, Medium, High
- Humidity into Dry, Normal, Humid

Binning Methods

- Quantile: Each bin will have approximately the same number of observations. If the Data tab includes a Weight variable, then the observations are weighted when performing the binning.
- KMeans: A kmeans clustering will be used for binning.
- Equal Width: The min to max range will be divided into equal width bins.
- We can also change the number of bins to be created for each method

Indicator variables

- Some model builders often do not directly handle categoric variables.
- One way to fix this is indicator or dummy variables
- For each level of a categoric variable, create a new indicator variable
- If for an observation the value of that categoric variable is that specific level, then the indicator value becomes 1, otherwise 0

Obs.	Colour	Colour_Red	Colour_Green	Colour_Blue
1	Green	0	1	0
2	Blue	0	0	1
3	Blue	0	0	1
4	Red	1	0	0
5	Green	0	1	0
6	Red	1	0	0

- Note that for a categoric variable with k unique levels, actually k-1 new variables will do. Why?

Join Categorics

Definition (Join Categorics)

The Join Categorics option provides a convenient way to stratify the dataset based on multiple categoric variables. It is a simple mechanism that creates a new variable from the combination of all of the values of the two constituent variables selected in the Rattle interface.

```
> ds$TJN <- interaction(paste(ds$RainToday, "_",
                              ds$RainTomorrow, sep=""))
> ds$TJN[grepl("^NA_|_NA$", ds$TJN)] <- NA
> ds$TJN <- as.factor(as.character(ds$TJN))
> head(ds[c("RainToday", "RainTomorrow", "TJN")])

  RainToday RainTomorrow     TJN
1        No          Yes  No_Yes
2       Yes          Yes Yes_Yes
3       Yes          Yes Yes_Yes
4       Yes          Yes Yes_Yes
5       Yes           No  Yes_No
6        No           No   No_No
```

Type Conversion

Definition (Type Conversion)

The As Categoric and As Numeric options will, respectively, convert a numeric variable to categoric and vice versa.

Example

```
> df$TFC_Cloud3pm <- as.factor(df$Cloud3pm)
> df$TNM_RainToday <- as.numeric(df$RainToday)
```

Cleanup

- It is quite easy to get our dataset variable count up to significant numbers.
- The Cleanup option allows us to tell Rattle to actually delete columns from the dataset.
- Options:
  - Delete Ignored: remove any variable that is ignored
  - Delete Selected: remove any variables we select
  - Delete Missing: remove any variables that have missing values
  - Delete Obs with Missing: remove observations (rather than variables) that have missing values.