Rsquared Academy

Practical Introduction to Market Basket Analysis
Association Rule Mining in R

Connect With Us

- Website (https://www.rsquaredacademy.com/)
- Free Online R Courses (https://rsquared-academy.thinkific.com/)
- R Packages (https://pkgs.rsquaredacademy.com)
- Shiny Apps (https://apps.rsquaredacademy.com)
- Blog (https://blog.rsquaredacademy.com)

- GitHub (https://github.com/rsquaredacademy)
- YouTube (https://www.youtube.com/user/rsquaredin/)
- Twitter (https://twitter.com/rsquaredacademy)
- Facebook (https://www.facebook.com/rsquaredacademy/)
- Linkedin (https://in.linkedin.com/company/rsquared-academy)

Resources

- Slides
- Code & Data
- RStudio Cloud

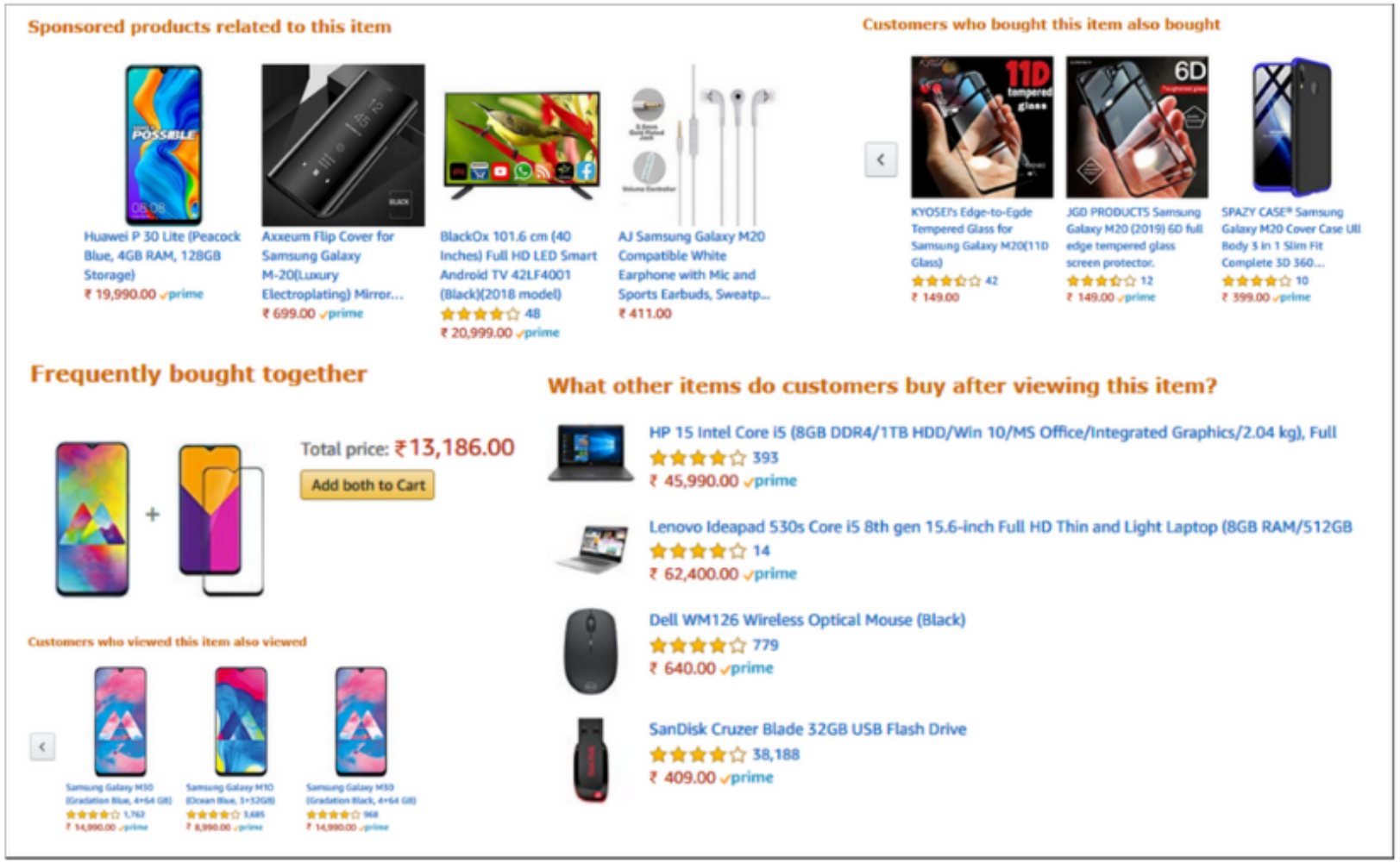
Agenda

- What?
- Why?
- How?

- Use Cases
- Demo

intro

Introduction



What

UNSUPERVISED DATA MINING TECHNIQUE

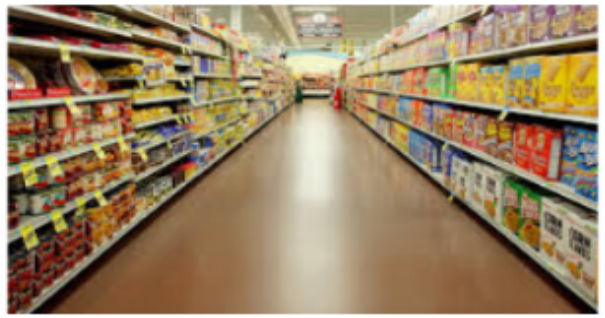
USED BY RETAILERS

TO IDENTIFY ITEMS FREQUENTLY BOUGHT TOGETHER

CREATES IF THEN SCENARIO RULES

Why

Store Layout



Recommendation Engines

Targeted Marketing

Up Sell & Cross Sell

Catalogue Design

Customer Experience

Advantages

Cost Effective

Insightful

Flexible

Actionable

Use Cases

Retail

Telecommunications

Banking

Medical

Manufacturing

Insurance

Simple Example

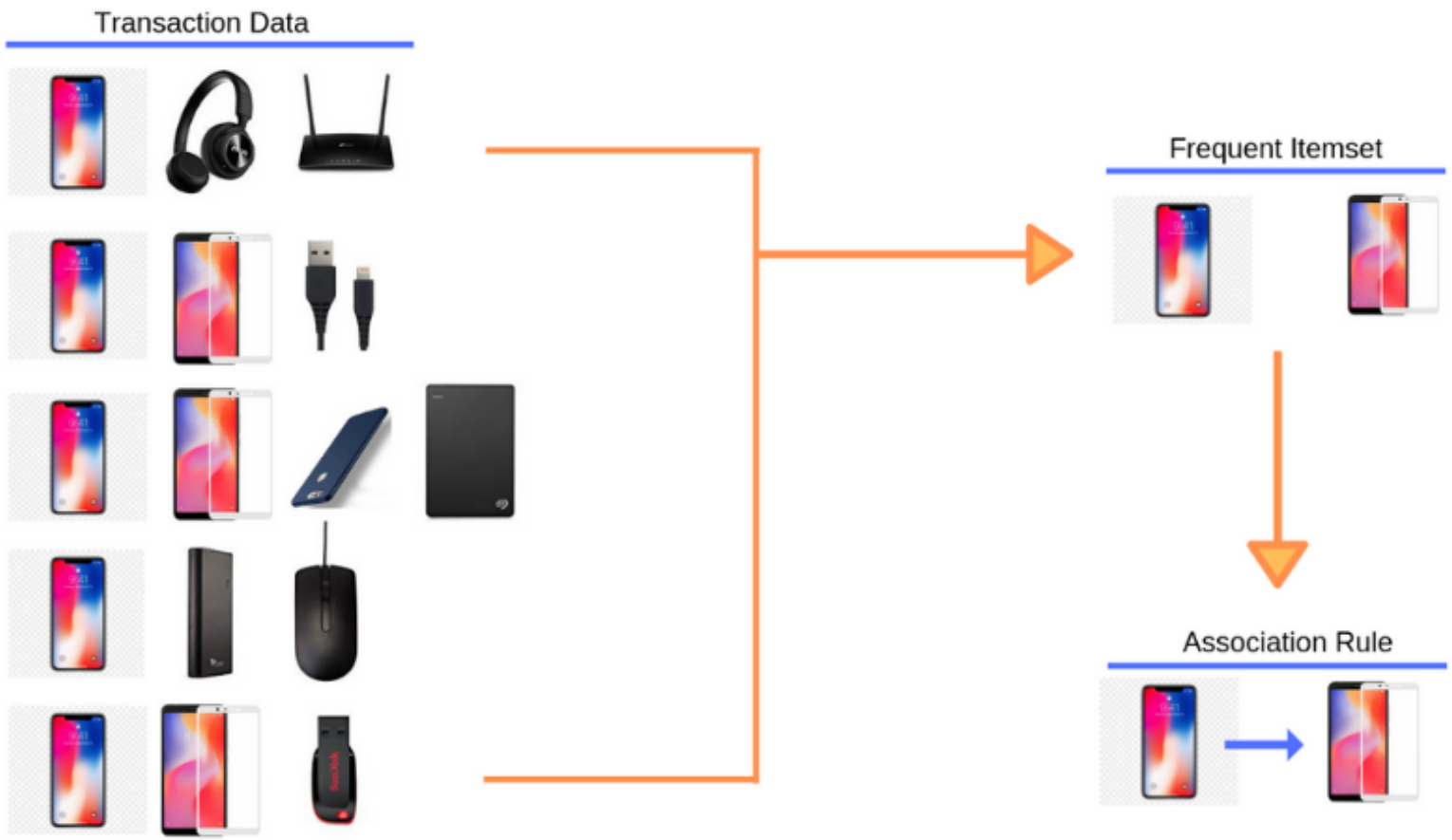




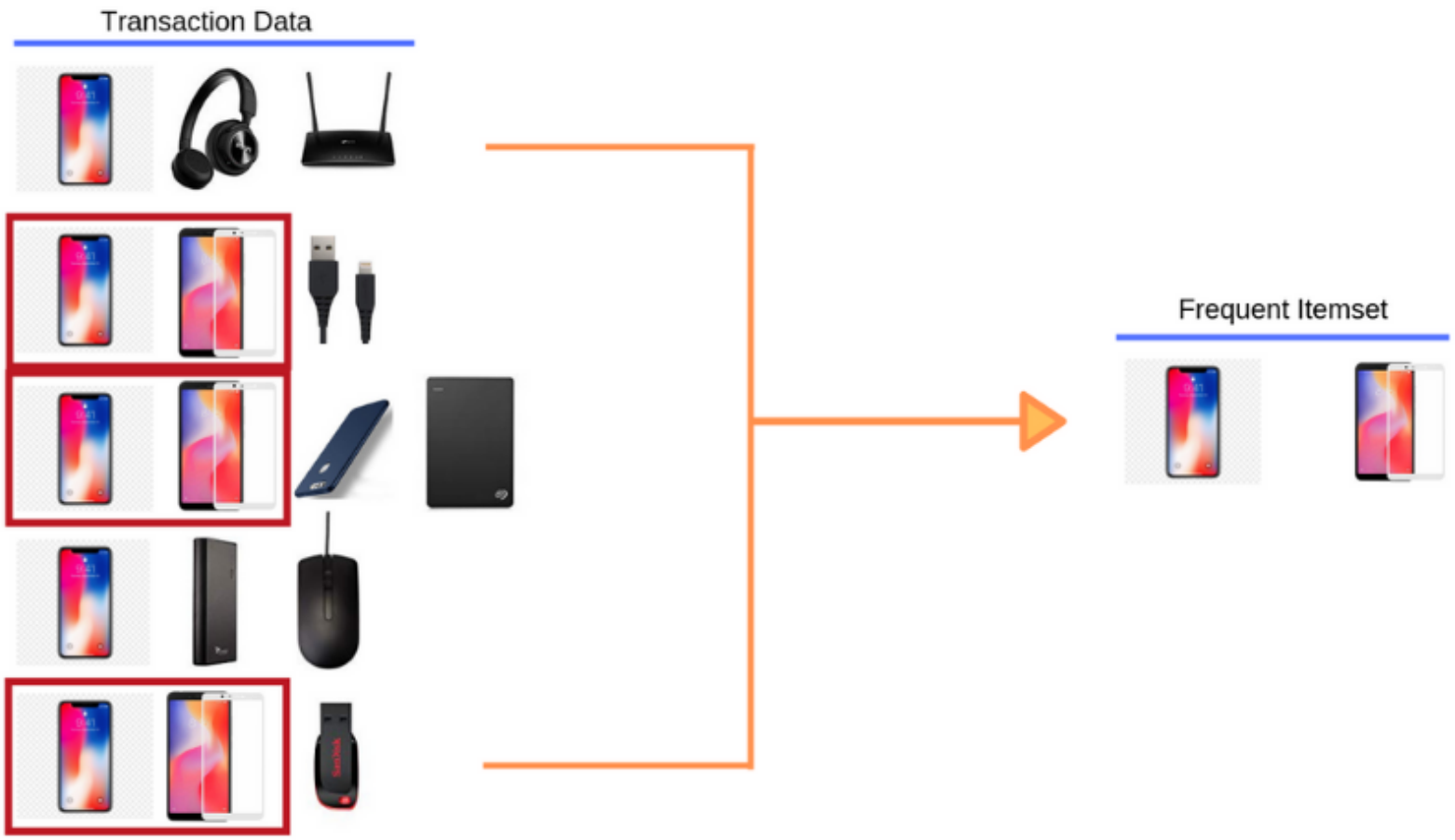




Steps



Itemset



Association Rule

Support

Association Rule

Mobile → Screen Guard

Total Transactions (N): 2000

Item	Transactions
Mobile	200
Screen Guard	160
Mobile + Screen Guard	120

$$\text{Support} = \frac{\text{Freq}(\text{Mobile} + \text{Screen Guard})}{\text{Total Transactions}} = \frac{120}{2000} = 0.06$$

Confidence

Association Rule

Mobile → Screen Guard

Total Transactions (N): 2000

Item	Transactions
Mobile	200
Screen Guard	160
Mobile + Screen Guard	120

$$\text{Confidence} = \frac{\text{Freq}(\text{Mobile} + \text{Screen Guard})}{\text{Freq}(\text{Mobile})} = \frac{120}{200} = 0.6$$

Lift

Association Rule

Mobile → Screen Guard

Total Transactions (N): 2000

Item	Transactions
Mobile	200
Screen Guard	160
Mobile + Screen Guard	120

$$\text{Lift} = \frac{\text{Support}\left(\text{Mobile} + \text{Screen Guard}\right)}{\text{Support}\left(\text{Mobile}\right) * \text{Support}\left(\text{Screen Guard}\right)} = \frac{0.06}{(0.1 * 0.08)} = 7.5$$

case studies

Data

- UCI
- data.world

Data Dictionary

- invoice number
- stock code
- description
- quantity
- invoice date
- unit price
- customer id
- country

Libraries

```
library(readxl)
library(readr)
library(mbar)
library(arules)
library(arulesViz)
library(magrittr)
library(dplyr)
library(lubridate)
library(forcats)
library(ggplot2)
```

Read Data

```
basket_data <- read.transactions("transaction_data.csv", format = "baske
  sep = ",")
basket_data
```

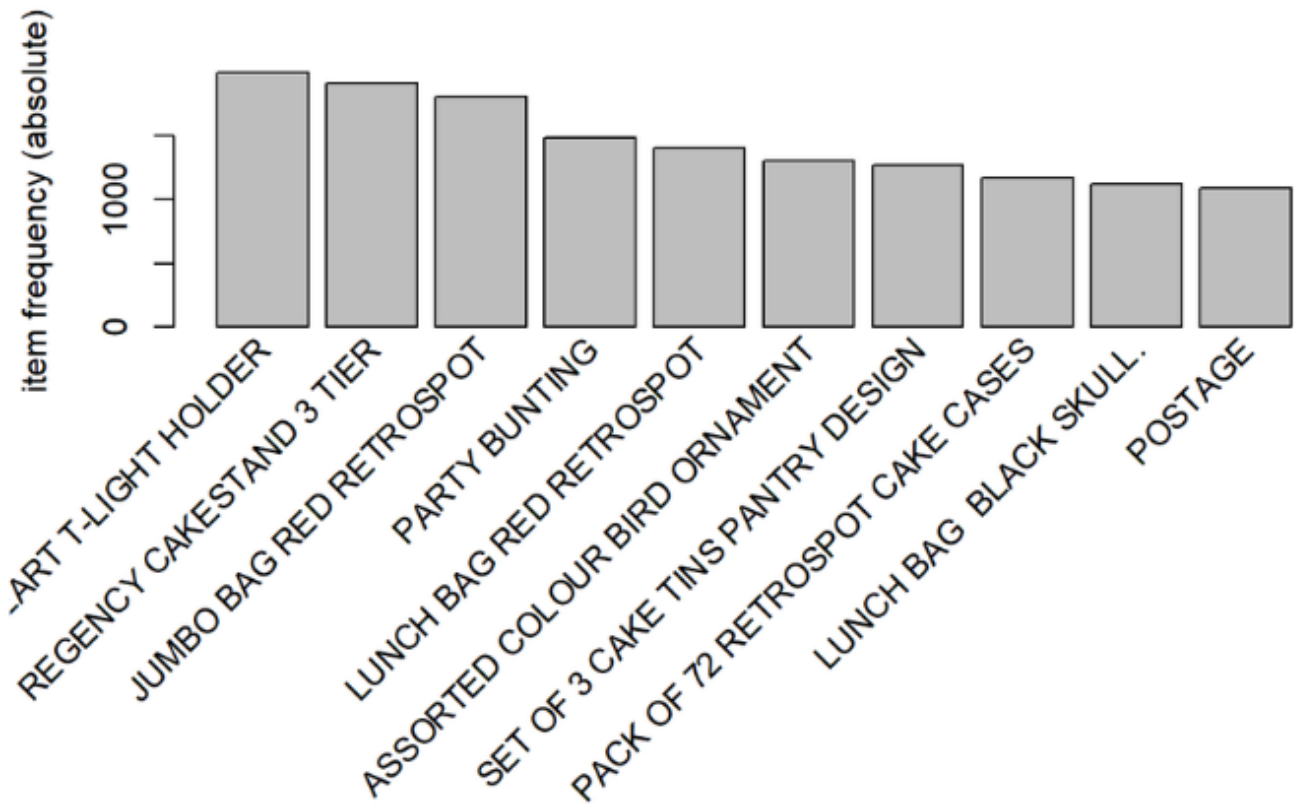
```
## transactions in sparse format with
##  25901 transactions (rows) and
##  10085 items (columns)
```

Data Summary

```
summary(basket_data)
```

```
## transactions as itemMatrix in sparse format with
##  25901 rows (elements/itemsets/transactions) and
##  10085 columns (items) and a density of 0.001660018
##
## most frequent items:
## WHITE HANGING HEART T-LIGHT HOLDER           REGENCY CAKESTAND 3 TIER
##                               1999                               1914
##            JUMBO BAG RED RETROSPOT                PARTY BUNTING
##                               1806                               1488
##            LUNCH BAG RED RETROSPOT                          (Other)
##                               1404                             425005
##
## element (itemset/transaction) length distribution:
## sizes
##    0    1    2    3    4    5    6    7    8    9   10   11   12   13
## 1454 4578 1727 1208  942  891  781  715  696  683  612  642  547  530
##   15   16   17   18   19   20   21   22   23   24   25   26   27   28
##  555  537  479  459  491  428  405  328  311  280  248  261  235  221
```

Item Frequency Plot



Generate Rules

```
rules <- apriori(basket_data, parameter = list(supp=0.009, conf=0.8,
    target = "rules", maxlen = 4))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support mi
##         0.8    0.1    1 none FALSE            TRUE       5   0.009

##  maxlen target   ext
##       4  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 233
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[10085 item(s), 25901 transaction(s)] done [1.41s
## sorting and recoding items     [508 item(s)] done [0.03s]
```

```
## Warning in apriori(basket_data, parameter = list(supp = 0.009, conf =
## 0.8, : Mining stopped (maxlen reached). Only patterns up to a length
## returned!
```

Rules Summary

```
summary(rules)
```

```
## set of 22 rules
##
## rule length distribution (lhs + rhs):sizes
##  2  3  4
## 11  9  2
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.000   2.000   2.500   2.591   3.000   4.000
##
## summary of quality measures:
##     support           confidence          lift           count
##  Min.   :0.009034   Min.   :0.8035   Min.   :22.59   Min.   :234.0
##  1st Qu.:0.010453   1st Qu.:0.8530   1st Qu.:25.02   1st Qu.:270.8
##  Median :0.013223   Median :0.8868   Median :55.94   Median :342.5
##  Mean   :0.012760   Mean   :0.9120   Mean   :48.55   Mean   :330.5
##  3rd Qu.:0.014362   3rd Qu.:1.0000   3rd Qu.:61.23   3rd Qu.:372.0
##  Max.   :0.018339   Max.   :1.0000   Max.   :71.30   Max.   :475.0
##
```

Inspect Rules

```
basket_rules <- sort(rules, by = 'confidence', decreasing = TRUE)
inspect(basket_rules[1:10])
```

```
##       lhs                                    rhs
## [1]   {BACK DOOR}                         => {KEY FOB}
## [2]   {SET 3 RETROSPOT TEA}               => {SUGAR}
## [3]   {SUGAR}                             => {SET 3 RETROSPOT TEA}
## [4]   {SET 3 RETROSPOT TEA}               => {COFFEE}

## [5]   {SUGAR}                             => {COFFEE}
## [6]   {SHED}                              => {KEY FOB}
## [7]   {SET 3 RETROSPOT TEA,
##        SUGAR}                             => {COFFEE}
## [8]   {COFFEE,
##        SET 3 RETROSPOT TEA}               => {SUGAR}
## [9]   {COFFEE,
##        SUGAR}                             => {SET 3 RETROSPOT TEA}
## [10] {PINK REGENCY TEACUP AND SAUCER,
##       REGENCY CAKESTAND 3 TIER,
##       ROSES REGENCY TEACUP AND SAUCER} => {GREEN REGENCY TEACUP AND S
```

Redundant Rules

rule	support	confidence	lift	count
{SET 3 RETROSPOT TEA, SUGAR} => {COFFEE}	0.01436238	1	55.94168	372
{SET 3 RETROSPOT TEA} => {COFFEE}	0.01436238	1	55.94168	372
{SUGAR} => {COFFEE}	0.01436238	1	55.94168	372

Redundant Rules

rule	support	confidence	lift	count
{COFFEE, SET 3 RETROSPOT TEA} => {SUGAR}	0.01436238	1	69.62634	372
{SET 3 RETROSPOT TEA} => {SUGAR}	0.01436238	1	69.62634	372
{COFFEE} => {SUGAR}	0.01436238	1	55.94168	372

Redundant Rules

rule	support	confidence	lift	count
{COFFEE, SUGAR} => {SET 3 RETROSPOT TEA}	0.01436238	1	69.62634	372
{COFFEE} => {SET 3 RETROSPOT TEA}	0.01436238	0.8034557	55.94168	372
{SUGAR} => {SET 3 RETROSPOT TEA}	0.01436238	1	69.62634	372

Redundant Rules

```
inspect(rules[is.redundant(rules)])
```

```
##     lhs                                rhs                    support
## [1] {SET 3 RETROSPOT TEA,SUGAR}  => {COFFEE}                0.01436238
## [2] {COFFEE,SET 3 RETROSPOT TEA} => {SUGAR}                 0.01436238
## [3] {COFFEE,SUGAR}               => {SET 3 RETROSPOT TEA} 0.01436238
##     confidence lift     count
## [1] 1          55.94168 372
## [2] 1          69.62634 372
## [3] 1          69.62634 372
```

Non Redundant Rules

```
inspect(rules[!is.redundant(rules)])
```

```
##       lhs                                          rhs
## [1]  {REGENCY TEA PLATE PINK}                   => {REGENCY TEA PLATE GREEN
## [2]  {BACK DOOR}                                => {KEY FOB}
## [3]  {SET 3 RETROSPOT TEA}                      => {SUGAR}
## [4]  {SUGAR}                                    => {SET 3 RETROSPOT TEA}
## [5]  {SET 3 RETROSPOT TEA}                      => {COFFEE}
## [6]  {COFFEE}                                   => {SET 3 RETROSPOT TEA}
## [7]  {SUGAR}                                    => {COFFEE}
## [8]  {COFFEE}                                   => {SUGAR}
## [9]  {REGENCY TEA PLATE GREEN}                  => {REGENCY TEA PLATE ROSES
## [10] {SHED}                                     => {KEY FOB}
## [11] {SET/6 RED SPOTTY PAPER CUPS}              => {SET/6 RED SPOTTY PAPER
## [12] {SET/20 RED RETROSPOT PAPER NAPKINS,
##       SET/6 RED SPOTTY PAPER CUPS}              => {SET/6 RED SPOTTY PAPER
## [13] {PINK REGENCY TEACUP AND SAUCER,
##       ROSES REGENCY TEACUP AND SAUCER}          => {GREEN REGENCY TEACUP AN
## [14] {GREEN REGENCY TEACUP AND SAUCER,
##       PINK REGENCY TEACUP AND SAUCER}           => {ROSES REGENCY TEACUP AN
```

What influenced purchase of product X?

```
sugar_rules <- apriori(basket_data, parameter = list(supp = 0.009, conf
  appearance = list(default = "lhs", rhs = "SUGAR"))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support mi
##         0.8    0.1    1 none FALSE            TRUE       5   0.009

##  maxlen target   ext
##      10  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 233
##
## set item appearances ...[1 item(s)] done [0.00s].
## set transactions ...[10085 item(s), 25901 transaction(s)] done [1.32s
## sorting and recoding items     [508 item(s)] done [0.03s]
```

What influenced purchase of product X?

```
rules_sugar <- sort(sugar_rules, by = "confidence", decreasing = TRUE)
inspect(rules_sugar)
```

```
##     lhs                                  rhs     support    confidence lif
## [1] {SET 3 RETROSPOT TEA}             => {SUGAR} 0.01436238 1.0000000  69.
## [2] {COFFEE,SET 3 RETROSPOT TEA}      => {SUGAR} 0.01436238 1.0000000  69.
## [3] {COFFEE}                          => {SUGAR} 0.01436238 0.8034557  55.
##     count
##
## [1] 372
## [2] 372
## [3] 372
```

What purchases did product X influence?

```
sugar_rules <- apriori(basket_data, parameter = list(supp = 0.009, conf
  appearance = list(default = "rhs", lhs = "SUGAR"))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support mi
##         0.8    0.1    1 none FALSE            TRUE       5   0.009

##  maxlen target   ext
##      10  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 233
##
## set item appearances ...[1 item(s)] done [0.00s].
## set transactions ...[10085 item(s), 25901 transaction(s)] done [1.35s
## sorting and recoding items     [508 item(s)] done [0.03s]
```

What purchases did product X influence?

```
rules_sugar <- sort(sugar_rules, by = "confidence", decreasing = TRUE)
inspect(rules_sugar)
```

```
##     lhs         rhs                     support    confidence lift     c
## [1] {SUGAR} => {SET 3 RETROSPOT TEA} 0.01436238 1          69.62634 3
## [2] {SUGAR} => {COFFEE}              0.01436238 1          55.94168 3
```

Top Rules by Support

```
supp_rules <- sort(rules, by = 'support', decreasing = TRUE)
top_rules <- supp_rules[1:10]
inspect(top_rules)
```

```
##       lhs                                       rhs
## [1]  {PINK REGENCY TEACUP AND SAUCER,
##       ROSES REGENCY TEACUP AND SAUCER} => {GREEN REGENCY TEACUP AND S
## [2]  {GREEN REGENCY TEACUP AND SAUCER,

##       PINK REGENCY TEACUP AND SAUCER}  => {ROSES REGENCY TEACUP AND S
## [3]  {SET 3 RETROSPOT TEA}             => {SUGAR}
## [4]  {SUGAR}                           => {SET 3 RETROSPOT TEA}
## [5]  {SET 3 RETROSPOT TEA}             => {COFFEE}
## [6]  {COFFEE}                          => {SET 3 RETROSPOT TEA}
## [7]  {SUGAR}                           => {COFFEE}
## [8]  {COFFEE}                          => {SUGAR}
## [9]  {SET 3 RETROSPOT TEA,
##       SUGAR}                           => {COFFEE}
## [10] {COFFEE,
##       SET 3 RETROSPOT TEA}             => {SUGAR}
```

Top Rules by Confidence

```
conf_rules <- sort(rules, by = 'confidence', decreasing = TRUE)
top_rules <- conf_rules[1:10]
inspect(top_rules)
```

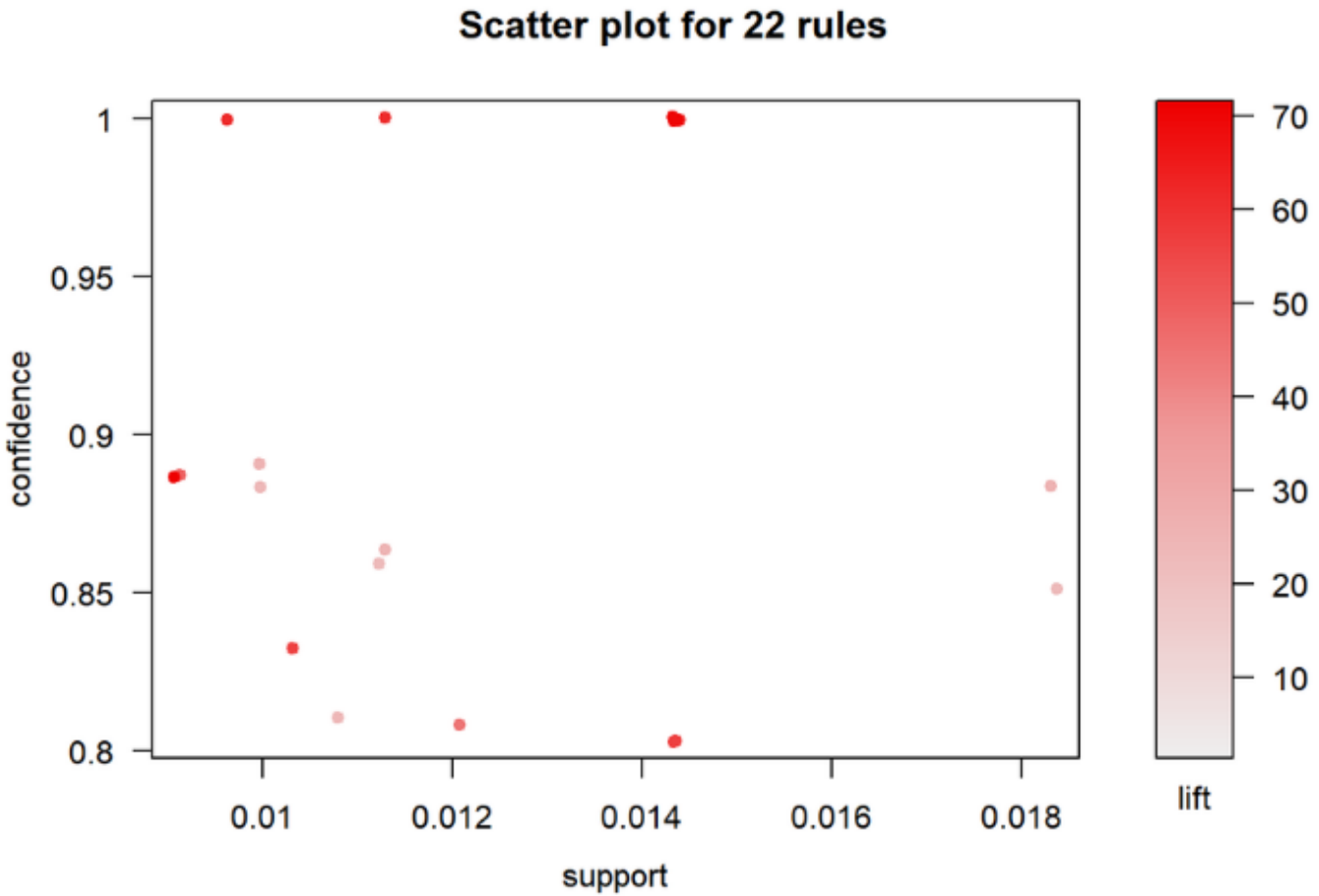
```
##       lhs                                    rhs
## [1]  {BACK DOOR}                          => {KEY FOB}
## [2]  {SET 3 RETROSPOT TEA}                => {SUGAR}
## [3]  {SUGAR}                              => {SET 3 RETROSPOT TEA}

## [4]  {SET 3 RETROSPOT TEA}                => {COFFEE}
## [5]  {SUGAR}                              => {COFFEE}
## [6]  {SHED}                               => {KEY FOB}
## [7]  {SET 3 RETROSPOT TEA,
##       SUGAR}                              => {COFFEE}
## [8]  {COFFEE,
##       SET 3 RETROSPOT TEA}                => {SUGAR}
## [9]  {COFFEE,
##       SUGAR}                              => {SET 3 RETROSPOT TEA}
## [10] {PINK REGENCY TEACUP AND SAUCER,
##       REGENCY CAKESTAND 3 TIER,
##       ROSES REGENCY TEACUP AND SAUCER} => {GREEN REGENCY TEACUP AND S
```

Top Rules by Lift

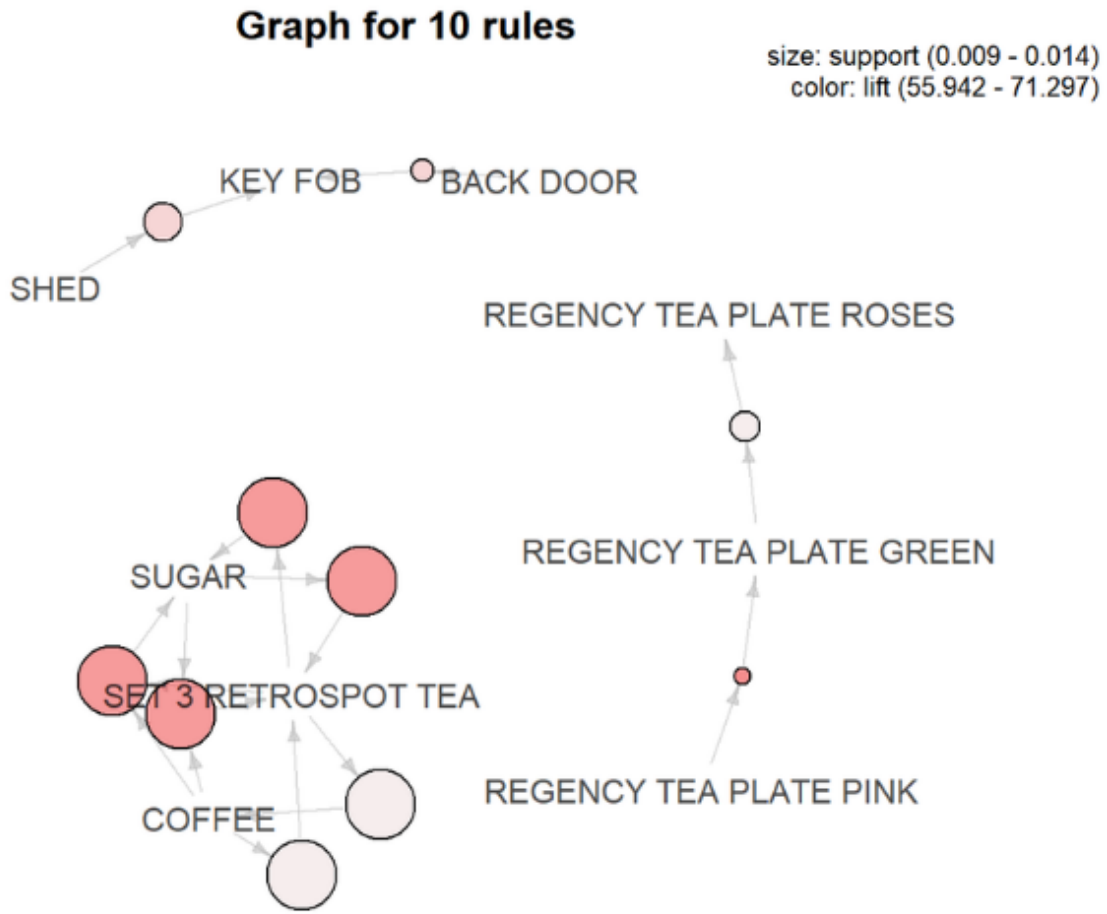
```
lift_rules <- sort(rules, by = 'lift', decreasing = TRUE)
top_rules <- lift_rules[1:10]
inspect(top_rules)
```

```
##      lhs                             rhs                              suppo
## [1]  {REGENCY TEA PLATE PINK}     => {REGENCY TEA PLATE GREEN} 0.0090344
## [2]  {SET 3 RETROSPOT TEA}        => {SUGAR}                   0.0143623
## [3]  {SUGAR}                      => {SET 3 RETROSPOT TEA}     0.0143623
## [4]  {COFFEE,
##       SET 3 RETROSPOT TEA}        => {SUGAR}                   0.0143623
## [5]  {COFFEE,
##       SUGAR}                      => {SET 3 RETROSPOT TEA}     0.0143623
## [6]  {BACK DOOR}                  => {KEY FOB}                 0.0096135
## [7]  {SHED}                       => {KEY FOB}                 0.0112736
## [8]  {REGENCY TEA PLATE GREEN}    => {REGENCY TEA PLATE ROSES} 0.0103470
## [9]  {SET 3 RETROSPOT TEA}        => {COFFEE}                  0.0143623
## [10] {COFFEE}                     => {SET 3 RETROSPOT TEA}     0.0143623
```

Scatter Plot



Network Plot



Directionality of rule is lost while using lift

Association Rule

Total Transactions (N): 2000

Item	Transactions
Mobile	200
Screen Guard	160
Mobile + Screen Guard	120

$$\text{Lift} = \frac{\text{Support}\left(\text{Mobile} + \text{Screen Guard}\right)}{\text{Support}\left(\text{Mobile}\right) * \text{Support}\left(\text{Screen Guard}\right)} = \frac{0.06}{(0.1 * 0.08)} = 7.5$$

Association Rule

Total Transactions (N): 2000

Item	Transactions
Mobile	200
Screen Guard	160
Mobile + Screen Guard	120

$$\text{Lift} = \frac{\text{Support}\left(\text{Screen Guard} + \text{Mobile}\right)}{\text{Support}\left(\text{Screen Guard}\right) * \text{Support}\left(\text{Mobile}\right)} = \frac{0.06}{(0.08 * 0.1)} = 7.5$$

Confidence as a measure can be misleading

Association Rule

Total Transactions (N): 2000

Item	Transactions
Mobile	200
Screen Guard	160
Mobile + Screen Guard	120

$$\text{Confidence} = \frac{\text{Freq}(\text{Mobile} + \text{Screen Guard})}{\text{Freq}(\text{Mobile})} = \frac{120}{200} = 0.6$$

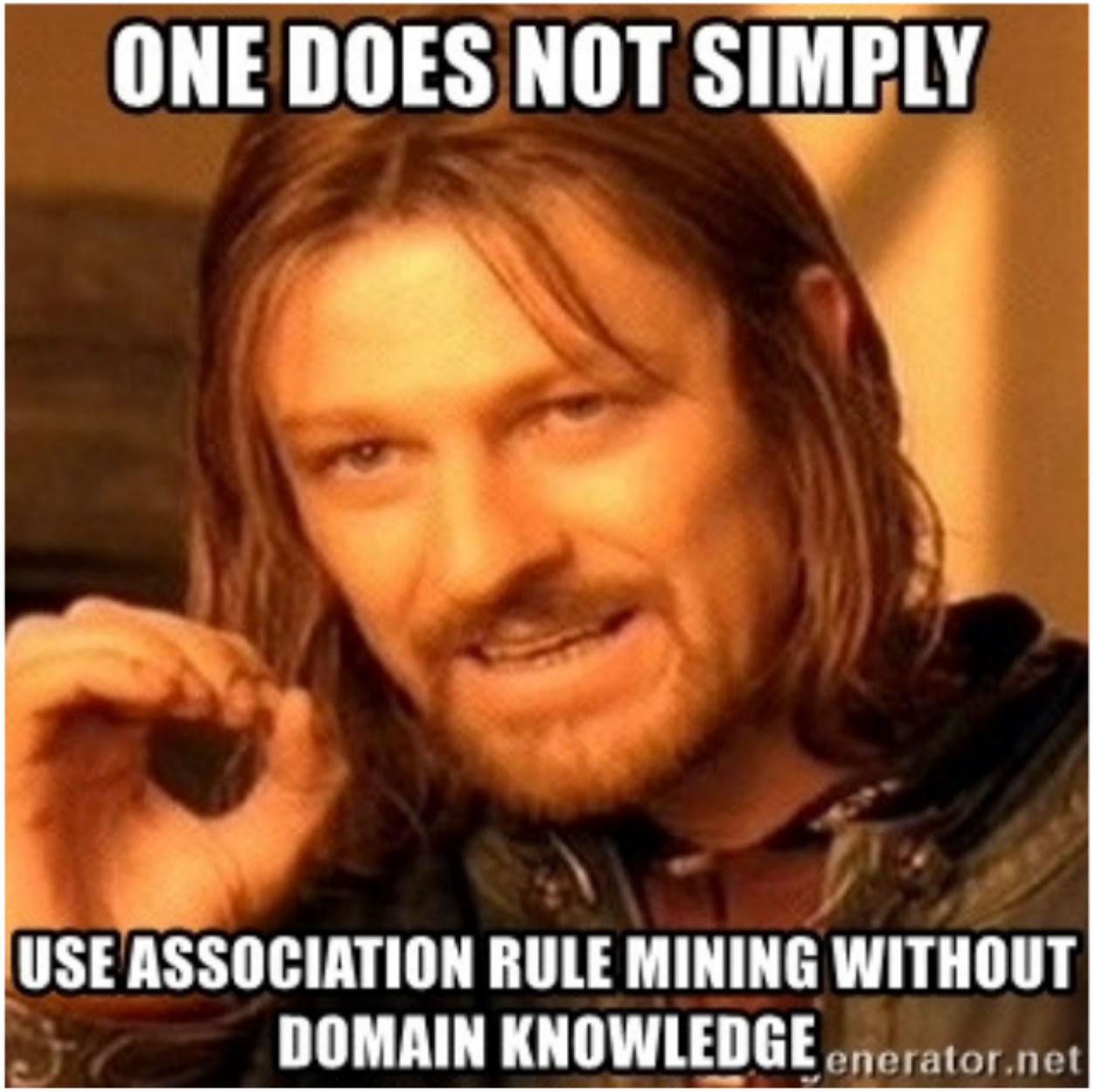
Association Rule

Total Transactions (N): 2000

Item	Transactions
Mobile	200
Screen Guard	160
Mobile + Screen Guard	120

$$\text{Confidence} = \frac{\text{Freq}(\text{Mobile} + \text{Screen Guard})}{\text{Freq}(\text{Screen Guard})} = \frac{120}{160} = \mathbf{0.75}$$

summary


ONE DOES NOT SIMPLY
USE ASSOCIATION RULE MINING WITHOUT DOMAIN KNOWLEDGE
enerator.net

Summary

- unsupervised data mining technique
- uncovers products frequently bought together
- creates if-then scenario rules
- cost-effective, insightful and actionable
- association rule mining has applications in several industries

- directionality of rule is lost while using lift
- confidence as a measure can be misleading

Thank You

For more information please visit our website
www.rsquaredacademy.com