Chapter 1: Getting Started with R and Machine Learning

```
Warning message:
In c(1, 3, 5, 7, 9) * c(2, 4) :
  longer object length is not a multiple of shorter object length
```

```
first second  third fourth  fifth
    1      2      3      4      5
```

```
second fourth
     2      4
```

```
, , first.set

     col1 col2 col3
row1    1    5    9
row2    2    6   10
row3    3    7   11
row4    4    8   12

, , second.set

     col1 col2 col3
row1   13   17   21
row2   14   18   22
row3   15   19   23
row4   16   20   24

, , third.set

     col1 col2 col3
row1   25   29    1
row2   26   30    2
row3   27   31    3
row4   28   32    4
```

```
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[2,]    5    6    7    8
[3,]    9   10   11   12
[4,]   13   14   15   16
[5,]   17   18   19   20
[6,]   21   22   23   24
```

```
[[1]]
[1] "row1" "row2" "row3" "row4"

[[2]]
[1] "col1" "col2" "col3"

[[3]]
[1] "first.set"  "second.set" "third.set"
```

```
[[1]]
[1] "r1" "r2" "r3" "r4" "r5" "r6"

[[2]]
[1] "c1" "c2" "c3" "c4"
```

```
   c1 c2 c3 c4
r1  1  2  3  4
r2  5  6  7  8
r3  9 10 11 12
r4 13 14 15 16
r5 17 18 19 20
r6 21 22 23 24
```

```
      M1.c1 M1.c2 M1.c3
M1.r1     1     2     3
M1.r2     4     5     6
M1.r3     7     8     9
M1.r4    10    11    12
M1.r5    13    14    15
```

```
      M2.c1 M2.c2 M2.c3
M2.r1    16    17    18
M2.r2    19    20    21
M2.r3    22    23    24
M2.r4    25    26    27
M2.r5    28    29    30
```

```
      M1.c1 M1.c2 M1.c3
M1.r1     1     2     3
M1.r2     4     5     6
M1.r3     7     8     9
M1.r4    10    11    12
M1.r5    13    14    15
M2.r1    16    17    18
M2.r2    19    20    21
M2.r3    22    23    24
M2.r4    25    26    27
M2.r5    28    29    30
```

```
      M1.c1 M1.c2 M1.c3 M2.c1 M2.c2 M2.c3
M1.r1     1     2     3    16    17    18
M1.r2     4     5     6    19    20    21
M1.r3     7     8     9    22    23    24
M1.r4    10    11    12    25    26    27
M1.r5    13    14    15    28    29    30
```

```
 [1]  1  4  7 10 13  2  5  8 11 14  3  6  9 12 15 16 19 22 25
[20] 28 17 20 23 26 29 18 21 24 27 30
```

```
      M1.c1 M1.c2 M1.c3
M1.r1    17    19    21
M1.r2    23    25    27
M1.r3    29    31    33
M1.r4    35    37    39
M1.r5    41    43    45
```

	M1.c1	M1.c2	M1.c3
M1.r1	16	34	54
M1.r2	76	100	126
M1.r3	154	184	216
M1.r4	250	286	324
M1.r5	364	406	450

	M2.r1	M2.r2	M2.r3	M2.r4	M2.r5
M2.c1	16	19	22	25	28
M2.c2	17	20	23	26	29
M2.c3	18	21	24	27	30

	M2.r1	M2.r2	M2.r3	M2.r4	M2.r5
M1.r1	104	122	140	158	176
M1.r2	257	302	347	392	437
M1.r3	410	482	554	626	698
M1.r4	563	662	761	860	959
M1.r5	716	842	968	1094	1220

	[,1]	[,2]	[,3]
[1,]	5	4	9
[2,]	-3	12	14
[3,]	2	-1	7

	[,1]	[,2]	[,3]
[1,]	0.19718310	-0.07444668	-0.1046278
[2,]	0.09859155	0.03420523	-0.1951710
[3,]	-0.04225352	0.02615694	0.1448692

	[,1]	[,2]	[,3]
[1,]	1	0	0
[2,]	0	1	0
[3,]	0	0	1

```
[[1]]
[1] 1 2 3 4 5

[[2]]
[1] "first"  "second" "third"

[[3]]
[1]  TRUE FALSE  TRUE  TRUE

[[4]]
function (x)  .Primitive("cos")

[[5]]
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
```

```
$even.nums
[1]  2  4  6  8 10

$odd.nums
[1] 1 3 5 7 9

$languages
[1] "R"      "Python" "Julia"  "Java"

$cosine.func
function (x)  .Primitive("cos")
```

```
$nums
[1] 1 2 3 4 5

$chars
[1] "a" "b" "c" "d" "e"

$cosine
function (x)  .Primitive("cos")

$languages
[1] "R"      "Python" "Java"

$months
[1] "Jan" "Feb" "Mar" "Apr"

$sine
function (x)  .Primitive("sin")
```

```
[[1]]
[1] 1

[[2]]
[1] 2

[[3]]
[1] 3

[[4]]
[1] 4

[[5]]
[1] 5
```

```
     real.name superhero.name franchise          team origin.year
1  Bruce Wayne         Batman        DC           JLA        1939
2   Clark Kent       Superman        DC           JLA        1938
3 Slade Wilson     Deathstroke       DC Suicide Squad        1980
4   Tony Stark       Iron Man    Marvel      Avengers        1963
5 Steve Rogers  Capt. America    Marvel      Avengers        1941
```

```
'data.frame':	5 obs. of  5 variables:
 $ real.name     : Factor w/ 5 levels "Bruce Wayne",..: 1 2 3 5 4
 $ superhero.name: Factor w/ 5 levels "Batman","Capt. America",..: 1 5 3 4 2
 $ franchise     : Factor w/ 2 levels "DC","Marvel": 1 1 1 2 2
 $ team          : Factor w/ 3 levels "Avengers","JLA",..: 2 2 3 1 1
 $ origin.year   : num  1939 1938 1980 1963 1941
```

```
[1] "real.name"      "superhero.name" "franchise"      "team"           "origin.year"
```

```
                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

```
    real.name superhero.name franchise          team origin.year
2  Clark Kent       Superman        DC           JLA        1938
3 Slade Wilson    Deathstroke        DC Suicide Squad        1980
4  Tony Stark       Iron Man    Marvel      Avengers        1963
```

```
     real.name superhero.name
2   Clark Kent       Superman
3 Slade Wilson    Deathstroke
4   Tony Stark       Iron Man
```

```
    real.name superhero.name franchise
1 Bruce Wayne         Batman        DC
2  Clark Kent       Superman        DC
```

```
     real.name superhero.name franchise
3 Slade Wilson    Deathstroke        DC
4   Tony Stark       Iron Man    Marvel
5 Steve Rogers  Capt. America    Marvel
```

```
      id         name     alias
1 emp001  Harvey Dent   TwoFace
2 emp003 Dick Grayson Nightwing
3 emp007   James Bond Agent 007
```

```
      id         name     alias     id     location      speciality
1 emp001  Harvey Dent   TwoFace emp001 Gotham City  Split Persona
2 emp003 Dick Grayson Nightwing emp003 Gotham City Expert Acrobat
3 emp007   James Bond Agent 007 emp007      London  Gadget Master
```

```
      id         name     alias    location      speciality
1 emp001  Harvey Dent   TwoFace Gotham City  Split Persona
2 emp003 Dick Grayson Nightwing Gotham City Expert Acrobat
3 emp007   James Bond Agent 007      London  Gadget Master
```

```
$l1
[1] 5.5

$l2
[1] 1010
```

```
$l1
 [1]  1  2  3  4  5  6  7  8  9 10

$l2
 [1] 0.3063285 0.3210605 0.2126607 0.4323474 0.7352608
 [6] 0.3211845 0.4266556 0.3350231 0.8402687 0.2214472

$l3
 [1] 3.163047 2.316373 2.928157 1.071683 1.961838 1.714548
 [7] 1.763979 3.798988 1.429736 2.898258
```

```
$l1
[1] 5.5

$l2
[1] 0.4152237

$l3
[1] 2.304661
```

```
       l1        l2        l3 
5.5000000 0.4152237 2.3046606 
```

```
          [,1]       [,2]        [,3]        [,4]
[1,] 0.1195527  0.7539491  1.04947756 -1.12405275
[2,] 0.1265696 -0.3927123 -0.13780092  0.07646778
[3,] 1.1871906  0.9269384  0.05736586  0.34318494
[4,] 0.6123884  1.7748904 -1.57002544 -0.53468646
[5,] 0.2013425 -1.5749354  0.45371789  0.29642974
```

```
          [,1]        [,2]      [,3]        [,4]       [,5]
25% -0.1913486 -0.20152876 0.2717302 -0.79352120 -0.2427270
50%  0.4367509 -0.03066657 0.6350617  0.03885096  0.2488861
75%  0.8278312  0.08899324 0.9920015  0.90301390  0.3357518
```

```
 [1]  1.00000000  2.00000000  3.00000000  4.00000000  5.00000000
 [6]  6.00000000  7.00000000  8.00000000  9.00000000 10.00000000
[11]  2.23147539  2.21731733  1.83956388  0.03597464  2.91214941
[16]  3.28026069  2.25403785  2.99538891  3.16527292  1.82685914
[21]  0.02740101  0.19610746  0.34837827  0.25190460  0.72999163
[26]  0.47645627  0.61436625  0.80770405  0.92255269  0.86156925
```

```
        1         2         3 
5.5000000 2.2758300 0.5236431 
```

```
$`1`
[1] 5.5

$`2`
[1] 2.27583

$`3`
[1] 0.5236431
```

```
$`1`
[1]  1 10

$`2`
[1] 0.03597464 3.28026069

$`3`
[1] 0.02740101 0.92255269
```

```
[[1]]
[1] 1 1 1 1

[[2]]
[1] 2 2 2

[[3]]
[1] 3 3

[[4]]
[1] 4
```

Statistical Data Analysis

Manuals

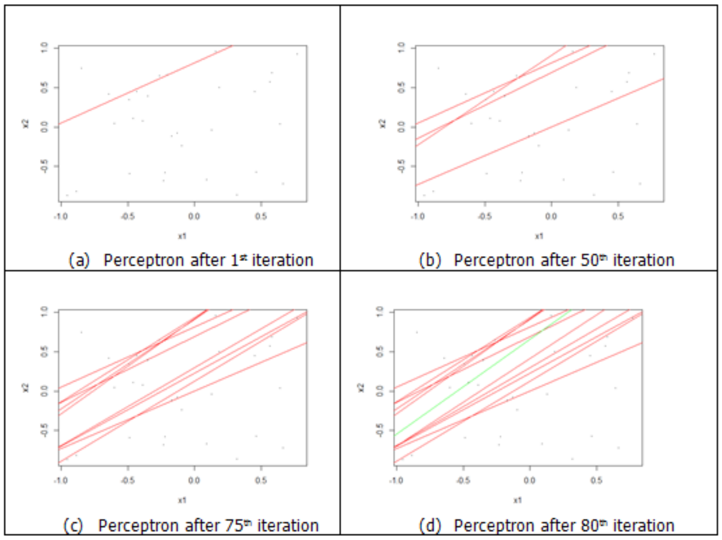
An Introduction to R	The R Language Definition
Writing R Extensions	R Installation and Administration
R Data Import/Export	R Internals

Reference

Packages	Search Engine & Keywords

Miscellaneous Material

About R	Authors	Resources
License	Frequently Asked Questions	Thanks
NEWS	User Manuals	Technical papers

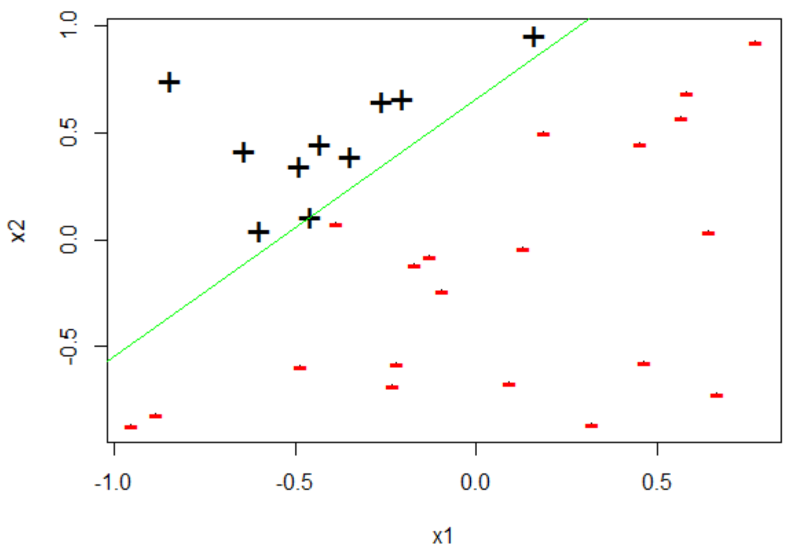


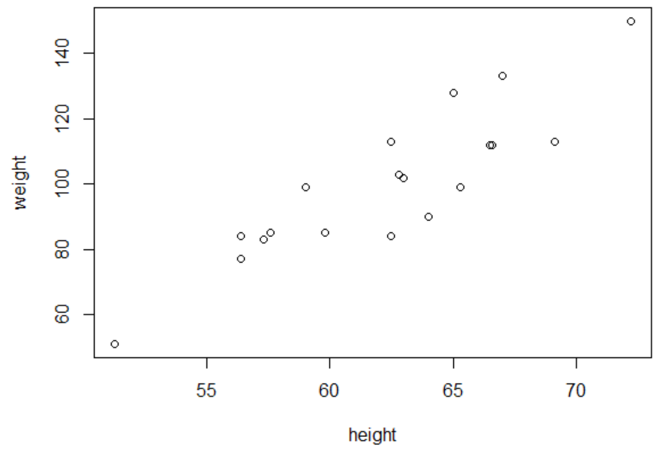
(a) Perceptron after 1st iteration

(b) Perceptron after 50th iteration

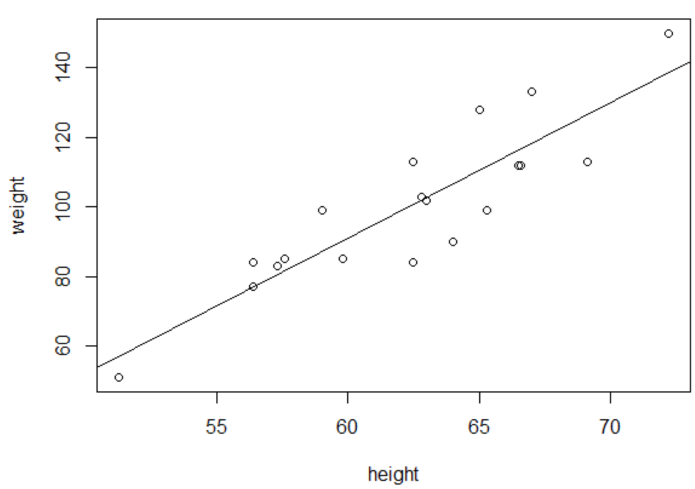
(c) Perceptron after 75th iteration

(d) Perceptron after 80th iteration





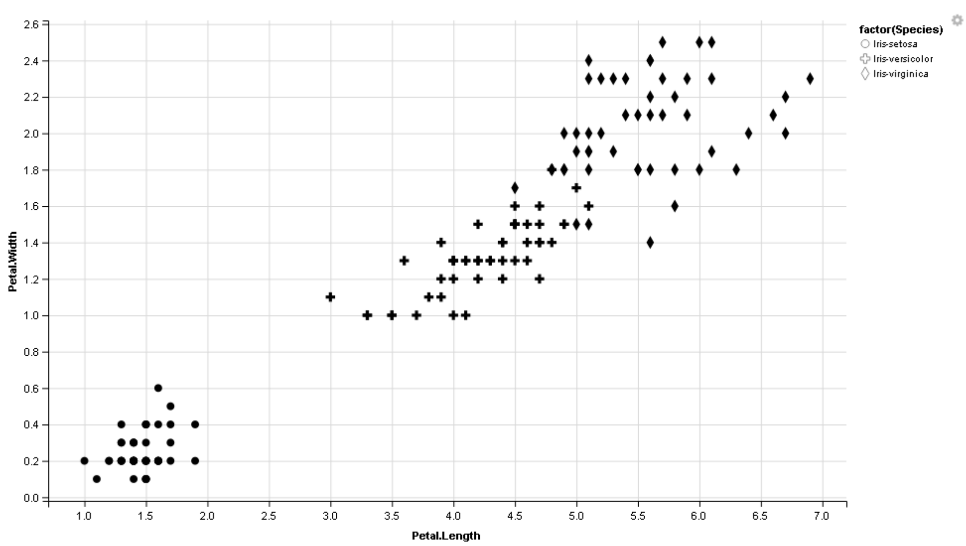

```
Call:
lm(formula = weight ~ height)

Coefficients:
(Intercept)       height
   -143.227        3.905
```



```
  Sepal.Length Sepal.Width Petal.Length Petal.Width     Species
1          5.1         3.5          1.4         0.2 Iris-setosa
2          4.9         3.0          1.4         0.2 Iris-setosa
3          4.7         3.2          1.3         0.2 Iris-setosa
4          4.6         3.1          1.5         0.2 Iris-setosa
5          5.0         3.6          1.4         0.2 Iris-setosa
6          5.4         3.9          1.7         0.4 Iris-setosa
```

```
'data.frame':   150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species     : Factor w/ 3 levels "Iris-setosa",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```
  Sepal.Length    Sepal.Width     Petal.Length    Petal.Width               Species
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100   Iris-setosa    :50
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300   Iris-versicolor:50
 Median :5.800   Median :3.000   Median :4.350   Median :1.300   Iris-virginica :50
 Mean   :5.843   Mean   :3.054   Mean   :3.759   Mean   :1.199
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
 Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
```




```
  Sepal.Length     Sepal.Width      Petal.Length     Petal.Width     
 Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.00000  
 1st Qu.:0.2222   1st Qu.:0.3333   1st Qu.:0.1017   1st Qu.:0.08333  
 Median :0.4167   Median :0.4167   Median :0.5678   Median :0.50000  
 Mean   :0.4287   Mean   :0.4392   Mean   :0.4676   Mean   :0.45778  
 3rd Qu.:0.5833   3rd Qu.:0.5417   3rd Qu.:0.6949   3rd Qu.:0.70833  
 Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.00000  
```

```
    Iris-setosa Iris-versicolor  Iris-virginica 
             50              50              50 
```

```
 [1] Iris-setosa     Iris-setosa     Iris-setosa     Iris-setosa    
 [5] Iris-setosa     Iris-setosa     Iris-setosa     Iris-setosa    
 [9] Iris-setosa     Iris-setosa     Iris-setosa     Iris-setosa    
[13] Iris-versicolor Iris-versicolor Iris-versicolor Iris-versicolor
[17] Iris-versicolor Iris-versicolor Iris-versicolor Iris-versicolor
[21] Iris-versicolor Iris-versicolor Iris-versicolor Iris-versicolor
[25] Iris-virginica  Iris-virginica  Iris-virginica  Iris-virginica 
[29] Iris-versicolor Iris-virginica  Iris-virginica  Iris-virginica 
[33] Iris-virginica  Iris-virginica  Iris-virginica  Iris-virginica 
[37] Iris-virginica  Iris-virginica  Iris-virginica  Iris-virginica 
Levels: Iris-setosa Iris-versicolor Iris-virginica
```

```
   Cell Contents
|-------------------------|
|                       N |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|

 
Total Observations in Table:  40 

 
                | iris_model 
iris.testLabels |     Iris-setosa | Iris-versicolor |  Iris-virginica |       Row Total | 
----------------|-----------------|-----------------|-----------------|-----------------|
    Iris-setosa |              12 |               0 |               0 |              12 | 
                |           1.000 |           0.000 |           0.000 |           0.300 | 
                |           1.000 |           0.000 |           0.000 |                 | 
                |           0.300 |           0.000 |           0.000 |                 | 
----------------|-----------------|-----------------|-----------------|-----------------|
Iris-versicolor |               0 |              12 |               0 |              12 | 
                |           0.000 |           1.000 |           0.000 |           0.300 | 
                |           0.000 |           0.923 |           0.000 |                 | 
                |           0.000 |           0.300 |           0.000 |                 | 
----------------|-----------------|-----------------|-----------------|-----------------|
 Iris-virginica |               0 |               1 |              15 |              16 | 
                |           0.000 |           0.062 |           0.938 |           0.400 | 
                |           0.000 |           0.077 |           1.000 |                 | 
                |           0.000 |           0.025 |           0.375 |                 | 
----------------|-----------------|-----------------|-----------------|-----------------|
   Column Total |              12 |              13 |              15 |              40 | 
                |           0.300 |           0.325 |           0.375 |                 | 
----------------|-----------------|-----------------|-----------------|-----------------|
```

$$\text{Apriori}(T, \varepsilon)$$

$$L_1 \leftarrow \{large1 - itemsets\}$$

$$k \leftarrow 2$$

$$\textbf{while } L_{k-1} \neq \theta$$

$$C_k \leftarrow \{a \cup \{b\} \mid a \in L_{k-1} \wedge b \notin a\} - \{c \mid \{s \mid s \subseteq c \wedge |s| = k-1\} \not\subset L_{k-1}\}$$

$$\textbf{for } \text{transactions } t \in T$$

$$C_t \leftarrow \{c \mid c \in C_k \wedge c \subseteq t\}$$

$$\textbf{for } \text{candidates } c \in C_t$$

$$count[c] \leftarrow count[c] + 1$$

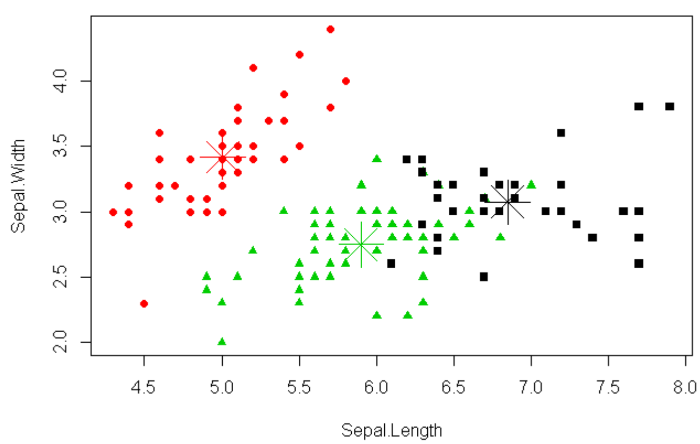
$$L_k \leftarrow \{c \mid c \in C_k \wedge count[c] \geq \varepsilon\}$$

$$k \leftarrow k + 1$$

$$\textbf{return} \bigcup_k L_k$$

```
> as(head(sort(rules, by = c("confidence", "support")), n=3), "data.frame")
                                                           rules   support confidence     lift
7                     {hours-per-week=Full-time} => {capital-loss=None} 0.5606650  0.9582531 1.005219
15                         {workclass=Private} => {capital-loss=None} 0.6639982  0.9564974 1.003377
50 {workclass=Private,native-country=United-States} => {capital-loss=None} 0.5897179  0.9554818 1.002312
```

```
                  1  2  3
Iris-setosa       0 50  0
Iris-versicolor   2  0 48
Iris-virginica   36  0 14
```



$$y = f\left(w_1x_1 + w_2x_2 + \ldots + w_nx_n + b\right) = f\left(w^Tx + b\right)$$

$$x_2 = x_1 + 1/2$$

$$\begin{aligned} y = {} & +1, \text{when } x_2 > x_1 + 1/2 \\ & -1 \text{ otherwise} \end{aligned}$$

$$y = b_0 + b_1x$$

$$residual_i = y_i - \hat{y}_i$$

$$\begin{aligned} &Sum\ of\ Squares\ of\ residual = SS\left(residual_i\right) \\ &SS\left(residual_i\right) = \sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2 \\ &\qquad\qquad = \sum_{i=1}^{n}\left(y_i - \left(b_0 + b_1x_i\right)\right)^2 \end{aligned}$$

$$Euclidean - distance\left(p, q\right) = \sqrt{\sum_{i=1}^{n}\left(q_i - p_i\right)^2}$$

$$\begin{aligned} &T = \left\{i_1, i_2, \ldots. i_n\right\} \\ &\text{and } T \subseteq {}_T \end{aligned}$$

$$x_{new} = \frac{x - \min\left(x\right)}{\max\left(x\right) - \min\left(x\right)}$$

$$
\begin{aligned}
&X \subseteq_T \\
&Y \subseteq_T \\
&X \cap Y = \phi
\end{aligned}
$$

$$
\begin{aligned}
&Set\ of\ n\ observations : \left\{x_1, x_2, \ldots . x_n\right\} \\
&Set\ S\ of\ partitions : S = \left\{S_1, S_2 \ldots S_k\right\} \\
&Objective\ is\ to\ minimize\ the\ with-in\ cluster\ sum\ of\ squares : \\
&\arg\min_s \sum_{x=1}^{k} \sum_{x \varepsilon S_i} || X - \mu_i ||
\end{aligned}
$$

	A	B	C	D
1	beer	diapers	bread	
2	diapers	eggs		
3	diapers	beer		
4	beer	diapers	eggs	
5	beer	diapers		
6	diapers	milk		
7	milk	bread		
8	diapers	beer	milk	bread
9	beer	diapers	milk	

$$S\left(IS_n\right)=\frac{f\left(IS_n\right)}{count\left(\sum_{i=1}^{n} IS_i\right)}$$

$$\frac{6}{9}$$

$$S\left(IS_x \rightarrow IS_y\right)=\frac{f\left(IS_x \cup IS_y\right)}{count\left(\sum_{i=1}^{n} IS_i\right)}$$

$$C\left(IS_x \rightarrow IS_y\right)=\frac{S\left(IS_x \cup IS_y\right)}{S\left(IS_x\right)}$$

$$C\left(IS_x \rightarrow IS_y\right)=\frac{f\left(IS_x \cup IS_y\right)}{f\left(IS_x\right)}$$

$$\frac{6}{6}$$

$$L\left(IS_x \to IS_y\right) = \frac{S\left(IS_x \cup IS_y\right)}{S\left(IS_x\right) \times S\left(IS_y\right)}$$

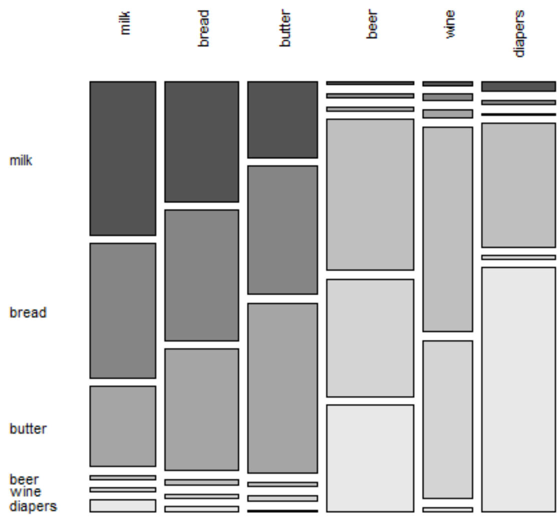
$$\frac{S\left(IS_{\{beer\}} \cup IS_{\{diapers\}}\right)}{S\left(IS_{\{beer\}}\right) \times S\left(IS_{\{diapers\}}\right)}$$

$$\frac{6 \div 9}{(6 \div 9) \times (8 \div 9)}$$

	A	B	C	D	E	F	G
1	Items	milk	bread	butter	beer	wine	diapers
2	milk	10000	8758	5241	300	215	753
3	bread	8758	9562	8865	427	322	353
4	butter	5241	8865	11753	310	447	114
5	beer	300	427	310	12985	10115	9173
6	wine	215	322	447	10115	7825	228
7	diapers	753	353	114	9173	228	18105

	milk	bread	butter	beer	wine	diapers
milk	10000	8758	5241	300	215	753
bread	8758	9562	8865	427	322	353
butter	5241	8865	11753	310	447	114
beer	300	427	310	12985	10115	9173
wine	215	322	447	10115	7825	228
diapers	753	353	114	9173	228	18105

Products Contingency Mosaic Plot



	whole milk	other vegetables	rolls/buns	soda	yogurt
whole milk	2513	736	557	394	551
other vegetables	736	1903	419	322	427
rolls/buns	557	419	1809	377	338
soda	394	322	377	1715	269
yogurt	551	427	338	269	1372

	whole milk	other vegetables	rolls/buns	soda	yogurt
whole milk	0.25551601	0.07483477	0.05663447	0.04006101	0.05602440
other vegetables	0.07483477	0.19349263	0.04260295	0.03274021	0.04341637
rolls/buns	0.05663447	0.04260295	0.18393493	0.03833249	0.03436706
soda	0.04006101	0.03274021	0.03833249	0.17437722	0.02735130
yogurt	0.05602440	0.04341637	0.03436706	0.02735130	0.13950178

	whole milk	other vegetables	rolls/buns	soda	yogurt
whole milk	NA	1.5136341	1.205032	0.8991124	1.571735
other vegetables	1.5136341	NA	1.197047	0.9703476	1.608457
rolls/buns	1.2050318	1.1970465	NA	1.1951242	1.339363
soda	0.8991124	0.9703476	1.195124	NA	1.124368
yogurt	1.5717351	1.6084566	1.339363	1.1243678	NA

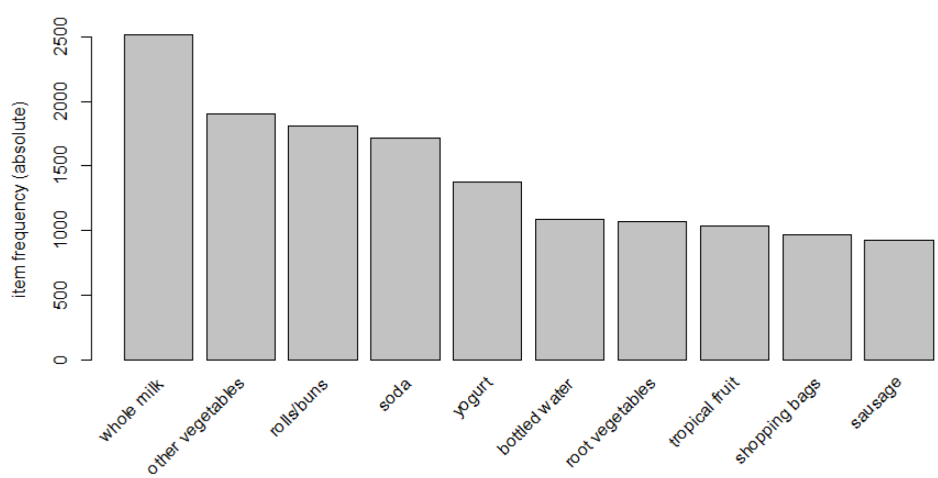
Item Association Matrix

	{beer, bread}	{beer, diapers}	{beer, milk}	{bread, diapers}	{bread, milk}	{diapers, milk}
{beer, diapers, bread}	1	1	NA	1	NA	NA
{diapers, eggs}	NA	NA	NA	NA	NA	NA
{diapers, beer}	NA	1	NA	NA	NA	NA
{beer, diapers, eggs}	NA	1	NA	NA	NA	NA
{beer, diapers}	NA	1	NA	NA	NA	NA
{diapers, milk}	NA	NA	NA	NA	NA	1
{milk, bread}	NA	NA	NA	NA	1	NA
{diapers, beer, milk, bread}	1	1	1	1	1	1
{beer, diapers, milk}	NA	1	1	NA	NA	1

	Itemset	Frequency	Support
1	{beer, diapers}	6	66.67
2	{diapers, milk}	3	33.33
3	{beer, bread}	2	22.22
4	{beer, milk}	2	22.22
5	{bread, diapers}	2	22.22
6	{bread, milk}	2	22.22

	Itemset	Frequency	Support
1	{beer, bread, diapers}	2	22.22
2	{beer, diapers, milk}	2	22.22

```
      whole milk other vegetables       rolls/buns             soda           yogurt
            2513             1903             1809             1715             1372
   bottled water  root vegetables   tropical fruit   shopping bags          sausage
            1087             1072             1032              969              924
```



	lhs		rhs	support	confidence	lift
1	{honey}	=>	{whole milk}	0.001118454	0.7333333	2.870009
2	{tidbits}	=>	{rolls/buns}	0.001220132	0.5217391	2.836542
3	{cocoa drinks}	=>	{whole milk}	0.001321810	0.5909091	2.312611
4	{pudding powder}	=>	{whole milk}	0.001321810	0.5652174	2.212062
5	{cooking chocolate}	=>	{whole milk}	0.001321810	0.5200000	2.035097

	lhs		rhs	support	confidence	lift
113	{rice,sugar}	=>	{whole milk}	0.001220132	1	3.913649
258	{canned fish,hygiene articles}	=>	{whole milk}	0.001118454	1	3.913649
1487	{root vegetables,butter,rice}	=>	{whole milk}	0.001016777	1	3.913649
1646	{root vegetables,whipped/sour cream,flour}	=>	{whole milk}	0.001728521	1	3.913649
1670	{butter,soft cheese,domestic eggs}	=>	{whole milk}	0.001016777	1	3.913649

	lhs		rhs	support	confidence	lift
53	{Instant food products,soda}	=>	{hamburger meat}	0.001220132	0.6315789	18.99565
37	{soda,popcorn}	=>	{salty snack}	0.001220132	0.6315789	16.69779
444	{flour,baking powder}	=>	{sugar}	0.001016777	0.5555556	16.40807
327	{ham,processed cheese}	=>	{white bread}	0.001931876	0.6333333	15.04549
330	{processed cheese,domestic eggs}	=>	{white bread}	0.001118454	0.5238095	12.44364

	lhs		rhs	support	confidence	lift
12	{coffee,misc. beverages}	=>	{soda}	0.001016777	0.7692308	4.411303
37	{sausage,bottled water,bottled beer}	=>	{soda}	0.001118454	0.7333333	4.205442
29	{sausage,white bread,shopping bags}	=>	{soda}	0.001016777	0.6666667	3.823129
34	{rolls/buns,bottled water,chocolate}	=>	{soda}	0.001321810	0.6500000	3.727551
13	{pastry,misc. beverages}	=>	{soda}	0.001220132	0.6315789	3.621912

	lhs		rhs	support	confidence	lift
8	{yogurt,sugar}	=>	{whole milk}	0.003660397	0.5294118	2.071932
2	{sugar}	=>	{whole milk}	0.015048297	0.4444444	1.739400
7	{yogurt,sugar}	=>	{other vegetables}	0.002846975	0.4117647	2.128064
4	{yogurt}	=>	{whole milk}	0.056024403	0.4016035	1.571735
1	{sugar}	=>	{other vegetables}	0.010777834	0.3183183	1.645119

Chapter 4: Building a Product Recommendation System

	Dell Inspiron	Mac Book Air	Acer Aspire	Alienware MX101	Mac Book Pro	Dell Vostro
User 1	?	5	2	?	4	1
User 2	4	?	3	?	1	4
User 3	3	?	?	5	4	3

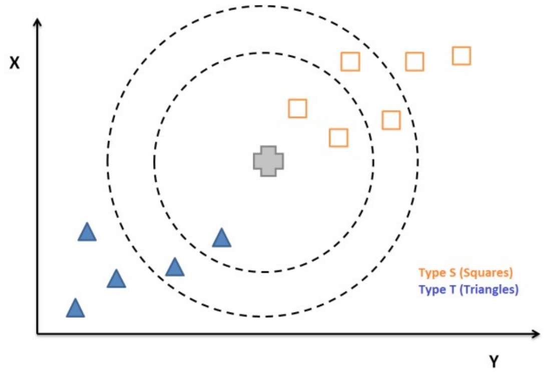
$$\subseteq$$

$$p_{ui} = \overline{r}_u + \frac{\sum_{u' \in N} s(u,u')\left(r_{u'i} - \overline{r}_u\right)}{\sum_{u' \in N} |\, s(u,u') \,|}$$

$$\overline{r}_u$$

$$s(u,u')$$

$$u'$$



$$s(u,v) = \frac{r_u . r_v}{\| r_u \|_2 \| r_v \|_2}$$

$$\| r_u \|_2$$

$$\| r_v \|_2$$

	iPhone.4	iPhone.5S	Nexus.5	Moto.X	Moto.G	Nexus.6	One.Plus.One
1	5	5	1	0	0	0	1
2	1	1	4	0	4	5	0
3	1	0	4	0	0	4	5
4	0	1	4	0	0	5	4
5	5	5	0	0	1	0	1
6	1	0	4	0	0	5	4
7	1	1	0	5	4	4	0
8	0	5	1	0	1	0	1

$$\hat{r}_{ij}$$

$$\begin{aligned} \hat{r}_{ij} &= x_i^T . y_i \\ &= \sum_{k=1}^{K} x_{ik} \cdot y_{kj} \end{aligned}$$

$$\begin{aligned} e_{ij}^2 &= \left(r_{ij} - \hat{r}_{ij} \right)^2 \\ &= \left(r_{ij} - \sum_{k=1}^{K} x_{ik} \cdot y_{kj} \right)^2 \end{aligned}$$

$$\frac{\partial \varepsilon_{ij}^2}{\partial x_{ik}} = -2\left(r_{ij} - \hat{r}_{ij}\right) \cdot y_{kj}$$
$$= -2e_{ij}y_{kj}$$
$$\frac{\partial \varepsilon_{ij}^2}{\partial y_{ik}} = -2\left(r_{ij} - \hat{r}_{ij}\right) . x_{ik}$$
$$= -2e_{ij}x_{ik}$$

$$x'_{ik} = x_{ik} + \alpha \frac{\partial e_{ij}^2}{\partial x_{ik}}$$
$$= x_{ik} + 2\alpha e_{ij} y_{kj}$$
$$y'_{kj} = y_{kj} + \alpha \frac{\partial e_{ij}^2}{\partial y_{ik}}$$
$$= y_{kj} + 2\alpha e_{ij} x_{ik}$$

$$x'_{ik}$$

$$y'_{kj}$$

$$e_{ij}^2 = \left(r_{ij} - \sum_{k=1}^{K} x_{ik} \cdot y_{kj} \right)^2 + \frac{\beta}{2} \sum_{k=1}^{K} \left(\| X \|^2 + \| Y \|^2 \right)$$

$$x'_{ik} = x_{ik} + \alpha \frac{\partial e_{ik}^2}{\partial x_{ik}}$$
$$= x_{ik} + 2\alpha e_{ij} y_{kj} - \beta x_{ik}$$

$$\begin{aligned} y'_{kj} &= y_{kj} + \alpha \frac{\partial e_{ij}^2}{\partial y_{ik}} \\ &= y_{kj} + 2\alpha e_{ij} x_{ik} - \beta y_{kj} \end{aligned}$$

	iPhone.4	iPhone.5s	Nexus.5	Moto.X	Moto.G	Nexus.6	One.Plus.One
1	4.97	4.99	1.00	3.41	1.01	2.05	0.99
2	1.06	0.99	4.03	5.58	4.20	4.74	4.38
3	0.87	0.80	3.97	5.43	4.14	4.64	4.32
4	1.13	1.06	3.97	5.53	4.14	4.69	4.31
5	4.97	4.99	1.00	3.41	1.00	2.05	0.99
6	1.06	1.00	3.97	5.51	4.14	4.68	4.32
7	1.01	0.95	3.58	4.98	3.73	4.22	3.88
8	4.94	4.96	1.00	3.39	1.00	2.04	0.99

iPhone.4	5
iPhone.5S	5
Nexus.5	1
Moto.X	0
Moto.G	0
Nexus.6	0
One.Plus.One	1

(a) Actual Ratings

iPhone.5s	4.99
iPhone.4	4.97
Moto.X	3.41
Nexus.6	2.05
Moto.G	1.01
Nexus.5	1.00
One.Plus.One	0.99

(b) Predicted Ratings

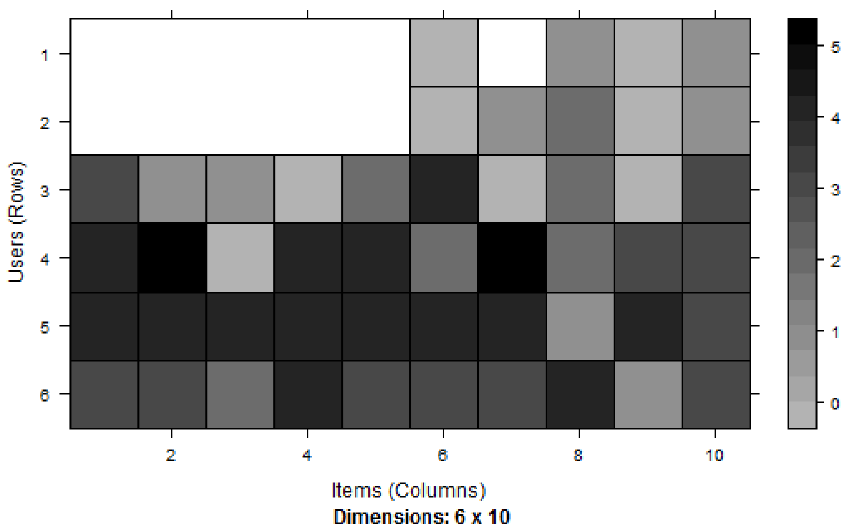
iPhone.4	1
iPhone.5S	0
Nexus.5	4
Moto.X	0
Moto.G	0
Nexus.6	4
One.Plus.One	5

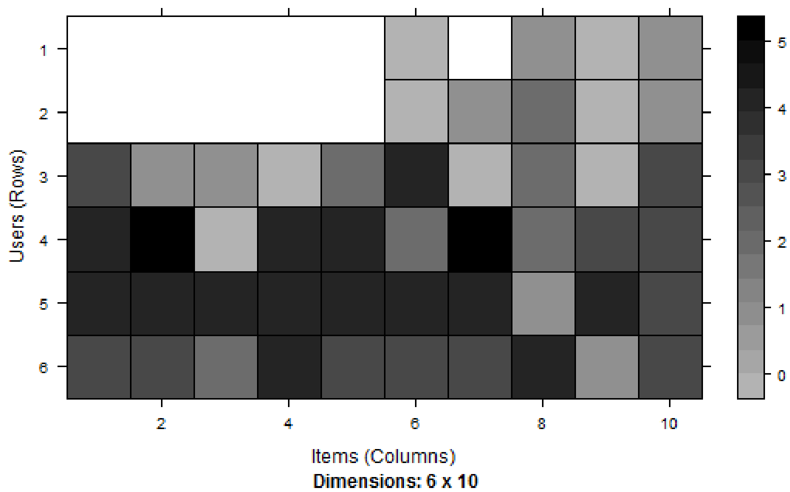
(a) Actual Ratings

iPhone.4	0.87
iPhone.5s	0.80
Nexus.5	3.97
Moto.X	5.43
Moto.G	4.14
Nexus.6	4.64
One.Plus.One	4.32

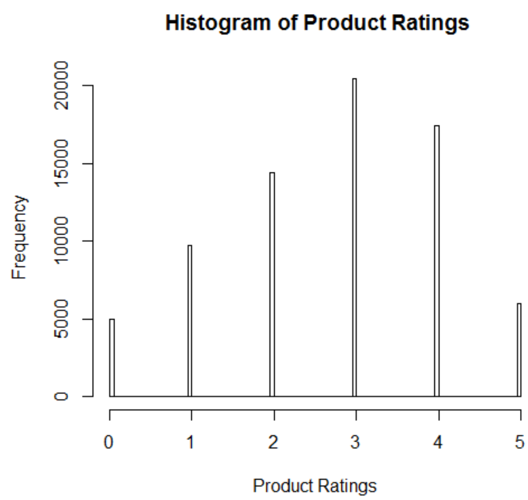
(b) Predicted Ratings

	UserID	Product	Ratings
1	u10005	prod_5	0
2	u10005	prod_7	0
3	u10005	prod_8	1
4	u10005	prod_13	0
5	u10005	prod_15	1
6	u10005	prod_16	0

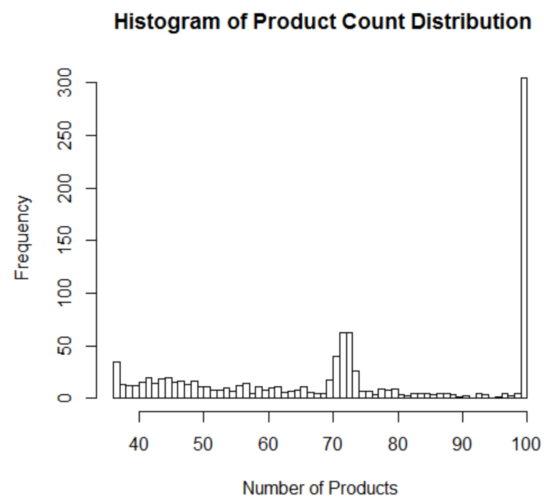




Users (Rows)
Items (Columns)
Dimensions: 6 x 10



Histogram of Product Ratings
Frequency
Product Ratings




```
       RMSE        MSE        MAE
1.1628422 1.3522021 0.9099283
```

```
                 RMSE      MSE       MAE
User Based CF 1.162842 1.352202 0.9099283
Item Based CF 1.221485 1.492025 0.9324538
```

$$r_{ij} = k, \text{if user has rated the item}; \text{where } k = 1 \text{ to } n$$
$$0, \text{otherwise}$$

$$N \subseteq U$$

$$R = U \times P^T \left(\text{we take transpose of } P \text{ as } P^T \text{ for matrix multiplication}\right)$$
$$\text{where}, |R| = |U| \times |P|$$

$$X = |U| \times K\ matrix$$
$$Y = |P| \times K\ matrix$$

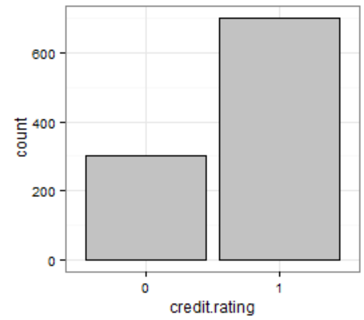
$$E = \Sigma\left(u_i, p_j, r_{ij}\right) \in S\, e_{ij}$$
$$= \Sigma\left(u_i, p_j, r_{ij}\right) \in S\left(r_{ij} - \sum_{k=1}^{K} x_{ik} y_{kj}\right)^2$$

```
  credit.rating account.balance credit.duration.months previous.credit.payment.status
1             1               1                     18                              4
2             1               1                      9                              4
3             1               2                     12                              2
4             1               1                     12                              4
5             1               1                     12                              4
6             1               1                     10                              4
  credit.purpose credit.amount savings employment.duration installment.rate
1              2          1049       1                   2                4
2              0          2799       1                   3                2
3              9           841       2                   4                2
4              0          2122       1                   3                3
5              0          2171       1                   3                4
6              0          2241       1                   2                1
  marital.status guarantor residence.duration current.assets age other.credits
1              2         1                  4              2  21             3
2              3         1                  2              1  36             3
3              2         1                  4              1  23             3
4              3         1                  2              1  39             3
5              3         1                  4              2  38             1
6              3         1                  3              1  48             3
  apartment.type bank.credits occupation dependents telephone foreign.worker
1              1            1          3          1         1              1
2              1            2          3          2         1              1
3              1            1          2          1         1              1
4              1            2          2          2         1              2
5              2            2          2          1         1              2
6              1            2          2          2         1              2
```

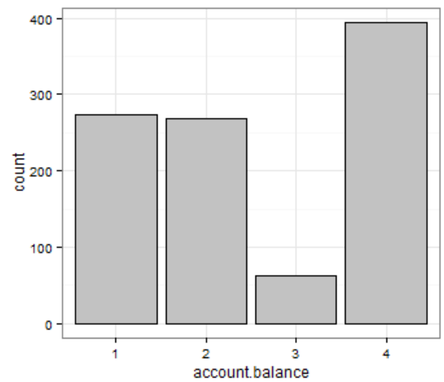
```
'data.frame':   1000 obs. of  21 variables:
 $ credit.rating                 : int  1 1 1 1 1 1 1 1 1 1 ...
 $ account.balance               : int  1 1 2 1 1 1 1 1 4 2 ...
 $ credit.duration.months        : int  18 9 12 12 12 10 8 6 18 24 ...
 $ previous.credit.payment.status: int  4 4 2 4 4 4 4 4 4 2 ...
 $ credit.purpose                : int  2 0 9 0 0 0 0 0 3 3 ...
 $ credit.amount                 : int  1049 2799 841 2122 2171 2241 3398..
 $ savings                       : int  1 1 2 1 1 1 1 1 1 3 ...
 $ employment.duration           : int  2 3 4 3 3 2 4 2 1 1 ...
 $ installment.rate              : int  4 2 2 3 4 1 1 2 4 1 ...
 $ marital.status                : int  2 3 2 3 3 3 3 3 2 2 ...
 $ guarantor                     : int  1 1 1 1 1 1 1 1 1 1 ...
 $ residence.duration            : int  4 2 4 2 4 3 4 4 4 4 ...
 $ current.assets                : int  2 1 1 1 2 1 1 1 3 4 ...
 $ age                           : int  21 36 23 39 38 48 39 40 65 23 ...
 $ other.credits                 : int  3 3 3 3 1 3 3 3 3 3 ...
 $ apartment.type                : int  1 1 1 1 2 1 2 2 2 1 ...
 $ bank.credits                  : int  1 2 1 2 2 2 2 1 2 1 ...
 $ occupation                    : int  3 3 2 2 2 2 2 2 1 1 ...
 $ dependents                    : int  1 2 1 2 1 2 1 2 1 1 ...
 $ telephone                     : int  1 1 1 1 1 1 1 1 1 1 ...
 $ foreign.worker                : int  1 1 1 2 2 2 2 2 1 1 ...
```

```
'data.frame':   1000 obs. of  21 variables:
 $ credit.rating                 : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
 $ account.balance               : Factor w/ 4 levels "1","2","3","4": 1 1 2 1 1 1 1 1 4 2 ...
 $ credit.duration.months        : int  18 9 12 12 12 10 8 6 18 24 ...
 $ previous.credit.payment.status: Factor w/ 5 levels "0","1","2","3",..: 5 5 3 5 5 5 5 5 5 3 ...
 $ credit.purpose                : Factor w/ 10 levels "0","1","2","3",..: 3 1 9 1 1 1 1 1 4 4 ...
 $ credit.amount                 : int  1049 2799 841 2122 2171 2241 3398 1361 1098 3758 ...
 $ savings                       : Factor w/ 5 levels "1","2","3","4",..: 1 1 2 1 1 1 1 1 1 3 ...
 $ employment.duration           : Factor w/ 5 levels "1","2","3","4",..: 2 3 4 3 3 2 4 2 1 1 ...
 $ installment.rate              : Factor w/ 4 levels "1","2","3","4": 4 2 2 3 4 1 1 2 4 1 ...
 $ marital.status                : Factor w/ 4 levels "1","2","3","4": 2 3 2 3 3 3 3 3 2 2 ...
 $ guarantor                     : Factor w/ 3 levels "1","2","3": 1 1 1 1 1 1 1 1 1 1 ...
 $ residence.duration            : Factor w/ 4 levels "1","2","3","4": 4 2 4 2 4 3 4 4 4 4 ...
 $ current.assets                : Factor w/ 4 levels "1","2","3","4": 2 1 1 1 2 1 1 1 3 4 ...
 $ age                           : int  21 36 23 39 38 48 39 40 65 23 ...
 $ other.credits                 : Factor w/ 3 levels "1","2","3": 3 3 3 3 1 3 3 3 3 3 ...
 $ apartment.type                : Factor w/ 3 levels "1","2","3": 1 1 1 1 2 1 2 2 2 1 ...
 $ bank.credits                  : Factor w/ 4 levels "1","2","3","4": 1 2 1 2 2 2 2 1 2 1 ...
 $ occupation                    : Factor w/ 4 levels "1","2","3","4": 3 3 2 2 2 2 2 2 1 1 ...
 $ dependents                    : Factor w/ 2 levels "1","2": 1 2 1 2 1 2 1 2 1 1 ...
 $ telephone                     : Factor w/ 2 levels "1","2": 1 1 1 1 1 1 1 1 1 1 ...
 $ foreign.worker                : Factor w/ 2 levels "1","2": 1 1 1 2 2 2 2 2 1 1 ...
```

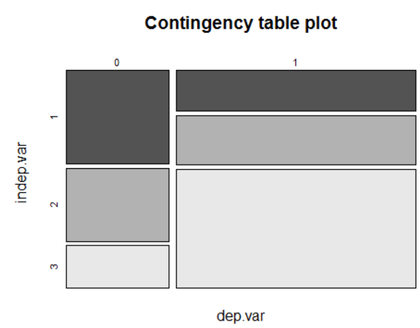
	credit.rating	Frequency	Proportion
2	1	700	0.7
1	0	300	0.3



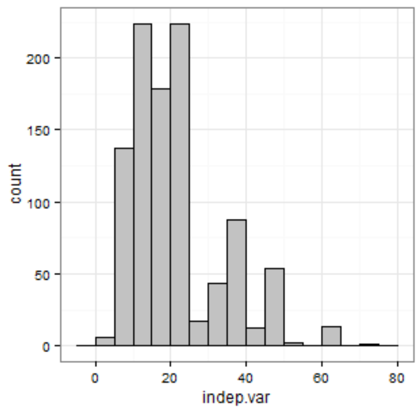
	account.balance	Frequency	Proportion
4	4	394	0.394
1	1	274	0.274
2	2	269	0.269
3	3	63	0.063

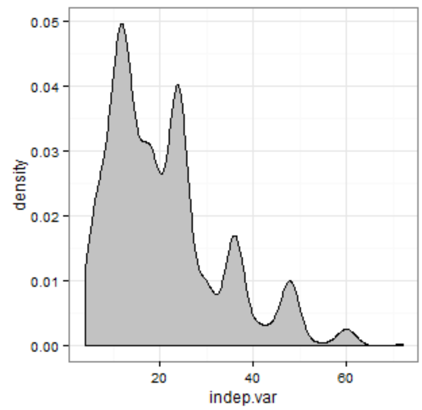


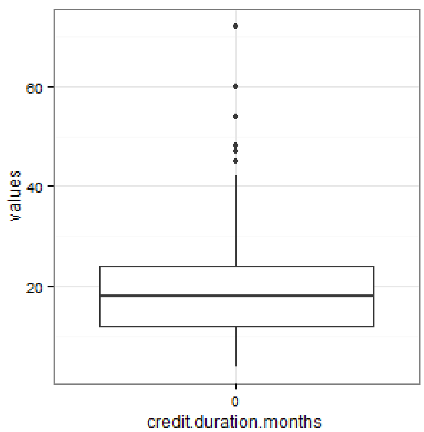

```
             | indep.var 
     dep.var |         1 |         2 |         3 | Row Total | 
-------------|-----------|-----------|-----------|-----------|
           0 |       135 |       105 |        60 |       300 | 
             |       0.5 |       0.4 |       0.1 |           | 
-------------|-----------|-----------|-----------|-----------|
           1 |       139 |       164 |       397 |       700 | 
             |       0.5 |       0.6 |       0.9 |           | 
-------------|-----------|-----------|-----------|-----------|
Column Total |       274 |       269 |       457 |      1000 | 
             |       0.3 |       0.3 |       0.5 |           | 
-------------|-----------|-----------|-----------|-----------|


Statistics for All Table Factors


Pearson's Chi-squared test 
------------------------------------------------------------
Chi^2 =  120.8438     d.f. =  2     p =  5.742621e-27 



Fisher's Exact Test for Count Data
------------------------------------------------------------
Alternative hypothesis: two.sided
p =  3.400743e-28 
```

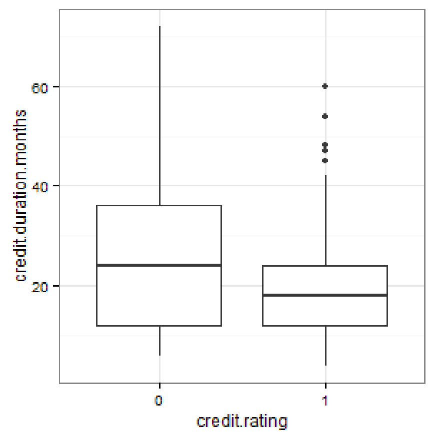


	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
credit.duration.months	4	12	18	20.9	24	72

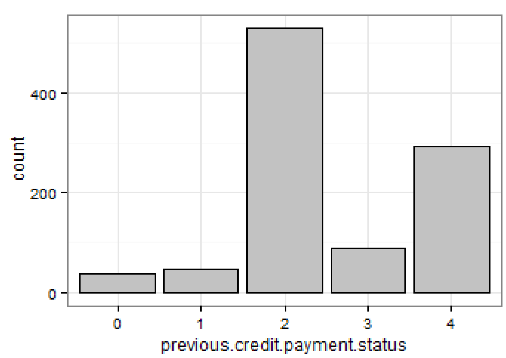








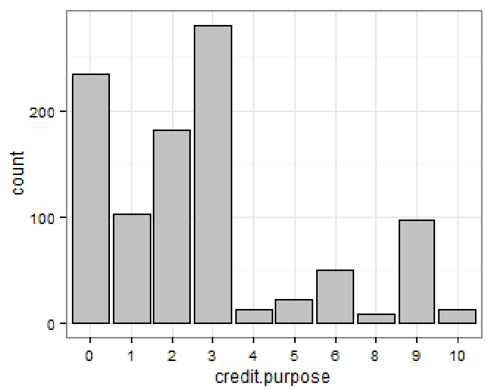
	previous.credit.payment.status	Frequency	Proportion
3	2	530	0.530
5	4	293	0.293
4	3	88	0.088
2	1	49	0.049
1	0	40	0.040



Total Observations in Table: 1000

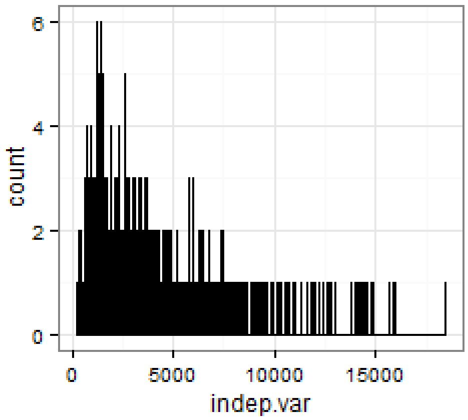
dep.var	indep.var 1	2	3	Row Total
0	53	169	78	300
	0.6	0.3	0.2	
1	36	361	303	700
	0.4	0.7	0.8	
Column Total	89	530	381	1000
	0.1	0.5	0.4	

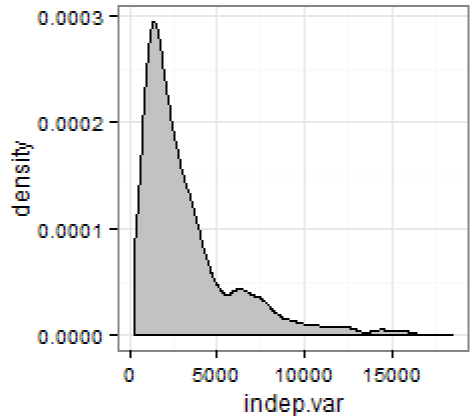
	credit.purpose	Frequency	Proportion
5	3	280	0.280
1	0	234	0.234
4	2	181	0.181
2	1	103	0.103
10	9	97	0.097
8	6	50	0.050
7	5	22	0.022
3	10	12	0.012
6	4	12	0.012
9	8	9	0.009

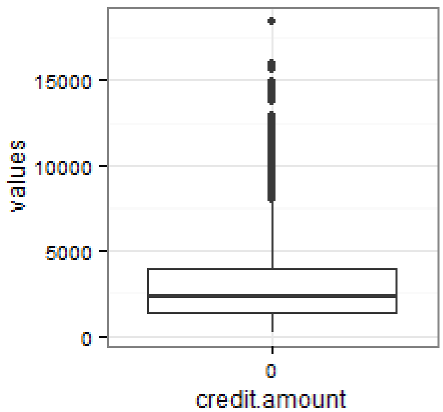


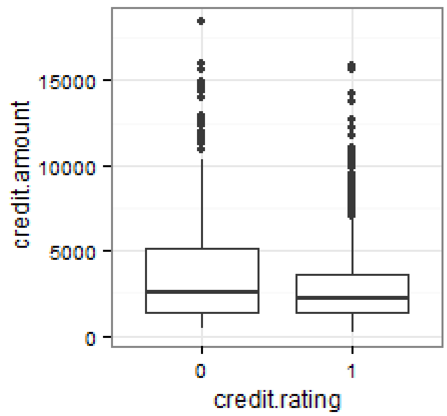
dep.var	indep.var 1	2	3	4	Row Total
0	17	58	96	129	300
	0.2	0.3	0.3	0.4	
1	86	123	268	223	700
	0.8	0.7	0.7	0.6	
Column Total	103	181	364	352	1000
	0.1	0.2	0.4	0.4	

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
credit.amount	250	1370	2320	3270	3970	18400









dep.var	indep.var 1	2	3	4	Row Total
0	217	34	17	32	300
	0.4	0.3	0.2	0.2	
1	386	69	94	151	700
	0.6	0.7	0.8	0.8	
Column Total	603	103	111	183	1000
	0.6	0.1	0.1	0.2	

dep.var	indep.var 1	2	3	4	Row Total
0	93	104	39	64	300
	0.4	0.3	0.2	0.3	
1	141	235	135	189	700
	0.6	0.7	0.8	0.7	
Column Total	234	339	174	253	1000
	0.2	0.3	0.2	0.3	

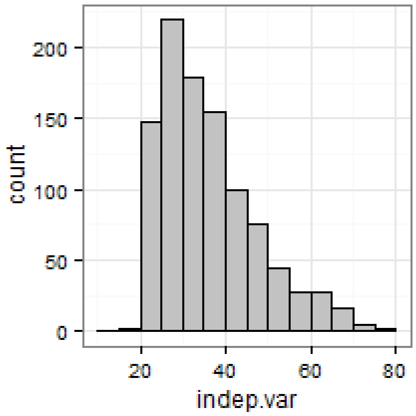
dep.var	indep.var 1	2	3	4	Row Total
0	34	62	45	159	300
	0.2	0.3	0.3	0.3	
1	102	169	112	317	700
	0.8	0.7	0.7	0.7	
Column Total	136	231	157	476	1000
	0.1	0.2	0.2	0.5	

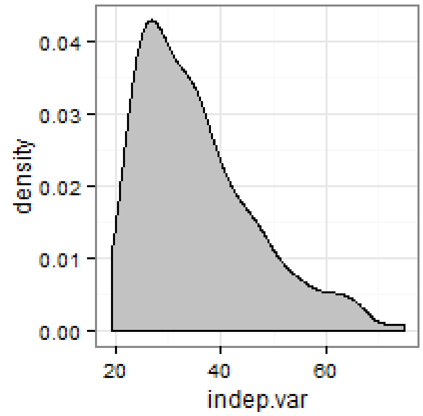
```
Statistics for All Table Factors

Pearson's Chi-squared test
------------------------------------------------------------
Chi^2 =  5.5     d.f. =  3     p =  0.14

Fisher's Exact Test for Count Data
------------------------------------------------------------
Alternative hypothesis: two.sided
p =  0.15
```

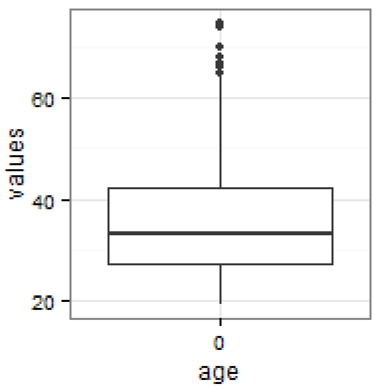
dep.var	indep.var 1	2	3	Row Total
0	129	146	25	300
	0.4	0.3	0.3	
1	231	402	67	700
	0.6	0.7	0.7	
Column Total	360	548	92	1000
	0.4	0.5	0.1	

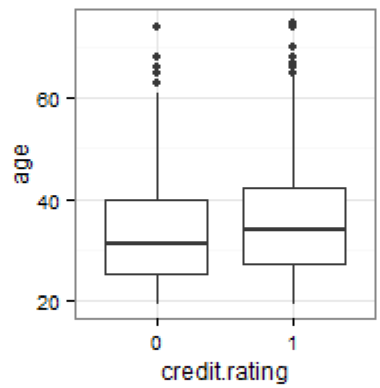
Fisher's Exact Test for Count Data	Pearson's Chi-squared test
data: credit.rating and residence.duration p-value = 0.9 alternative hypothesis: two.sided	data: credit.rating and residence.duration X-squared = 0.7, df = 3, p-value = 0.9

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
age	19	27	33	35.5	42	75


200
150
100
50
0
count
20
40
60
80
indep.var

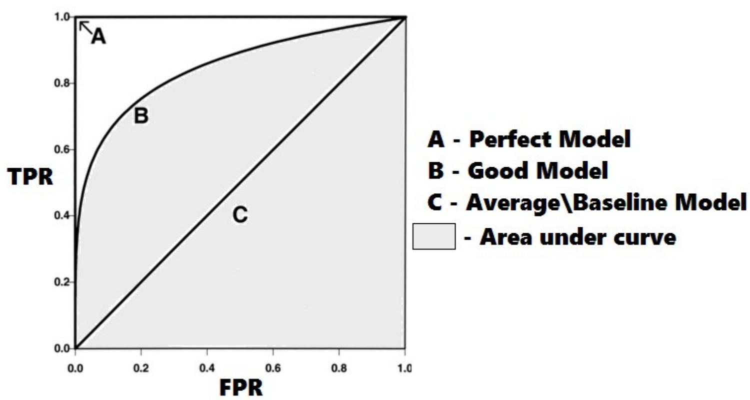

0.04
0.03
0.02
0.01
0.00
density
20
40
60
indep.var


60
40
20
values
0
age


60
40
20
age
0
1
credit.rating

Chapter 6: Credit Risk Detection and Prediction – Predictive Analytics

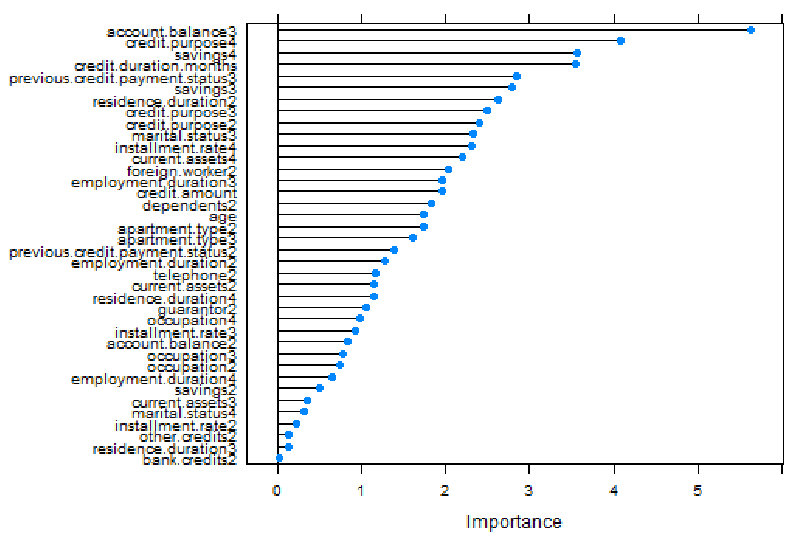
	Actual Labels	
Predicted Labels	0	1
0	22	20
1	18	40
Specificity (TNR):	22 / 40 = 0.55	
Sensitivity (TPR):	40 / 60 = 0.67	
Precision (PPV):	40 / 58 = 0.69	
NPV:	22 / 42 = 0.52	
FPR (1-Specificity):	18 / 40 = 0.45	
FNR (1- Sensitivity):	20 / 60 = 0.33	
Accuracy:	62 / 100 = 0.62	
F1 Score:	80 / 118 = 0.68	



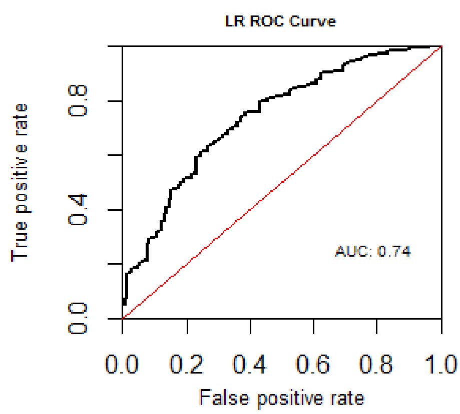

```
Recursive feature selection

Outer resampling method: Cross-Validated (20 fold)

Resampling performance over subset size:

 Variables Accuracy  Kappa AccuracySD KappaSD Selected
         1   0.7167 0.0000    0.01565  0.0000
         2   0.7584 0.2892    0.06599  0.2042
         3   0.7485 0.2467    0.06253  0.2006
         4   0.7551 0.3538    0.07602  0.1877
         5   0.7838 0.4382    0.07132  0.1712
         6   0.7869 0.4297    0.06066  0.1551
         7   0.7768 0.3915    0.04241  0.1220
         8   0.7868 0.4047    0.03559  0.1163
         9   0.7801 0.3997    0.05498  0.1517
        10   0.7869 0.4170    0.05725  0.1665        *
        20   0.7567 0.2858    0.05634  0.1856

The top 5 variables (out of 10):
   account.balance, credit.duration.months, savings, previous.credit.payment.status, credit.amount
```

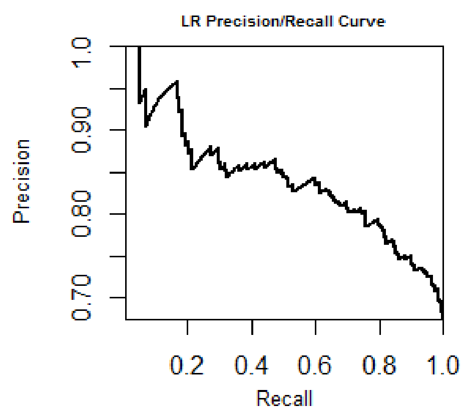
```
Call:
glm(formula = formula.init, family = "binomial", data = train.data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.6551  -0.5141   0.3187   0.6335   2.1838

Coefficients:
                                 Estimate Std. Error z value Pr(>|z|)
(Intercept)                      0.156837   1.073803   0.146 0.883876
account.balance2                 0.246170   0.291234   0.845 0.397963
account.balance3                 1.678654   0.298253   5.628 1.82e-08 ***
credit.duration.months          -0.545377   0.153737  -3.547 0.000389 ***
previous.credit.payment.status2  0.588709   0.420863   1.399 0.161869
previous.credit.payment.status3  1.225405   0.431435   2.840 0.004507 **
```

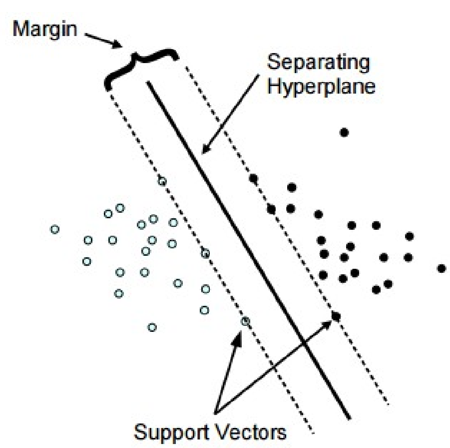
```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0  62  45
         1  68 225

               Accuracy : 0.7175
                 95% CI : (0.6706, 0.7611)
    No Information Rate : 0.675
    P-Value [Acc > NIR] : 0.03788

                  Kappa : 0.3252
 Mcnemar's Test P-Value : 0.03849

            Sensitivity : 0.8333
            Specificity : 0.4769
         Pos Pred Value : 0.7679
         Neg Pred Value : 0.5794
             Prevalence : 0.6750
         Detection Rate : 0.5625
   Detection Prevalence : 0.7325
      Balanced Accuracy : 0.6551

       'Positive' Class : 1
```



```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0  35  16
         1  95 254

               Accuracy : 0.7225
                 95% CI : (0.6758, 0.7658)
    No Information Rate : 0.675
    P-Value [Acc > NIR] : 0.02302

                  Kappa : 0.2492
 Mcnemar's Test P-Value : 1.327e-13

            Sensitivity : 0.9407
            Specificity : 0.2692
         Pos Pred Value : 0.7278
         Neg Pred Value : 0.6863
             Prevalence : 0.6750
         Detection Rate : 0.6350
   Detection Prevalence : 0.8725
      Balanced Accuracy : 0.6050

       'Positive' Class : 1
```

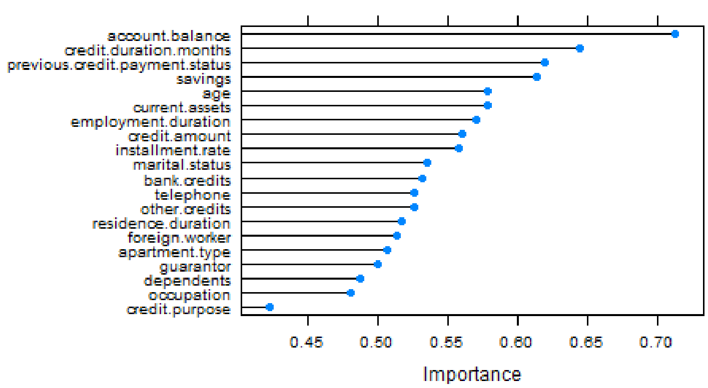






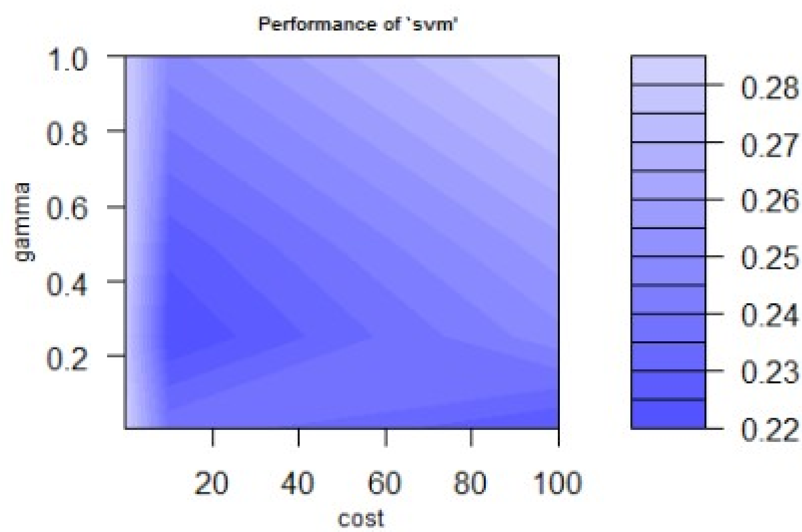
```
Call:
svm(formula = formula.init, data = train.data, kernel = "radial", cost = 100, gamma = 1)

Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  radial
       cost:  100
      gamma:  1

Number of Support Vectors:  600

 ( 430 170 )

Number of Classes:  2

Levels:
 0 1
```

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0   0   0
         1 130 270

               Accuracy : 0.675
                 95% CI : (0.6267, 0.7207)
    No Information Rate : 0.675
    P-Value [Acc > NIR] : 0.5238

                  Kappa : 0
 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 1.000
            Specificity : 0.000
         Pos Pred Value : 0.675
         Neg Pred Value :   NaN
             Prevalence : 0.675
         Detection Rate : 0.675
   Detection Prevalence : 1.000
      Balanced Accuracy : 0.500

       'Positive' Class : 1
```

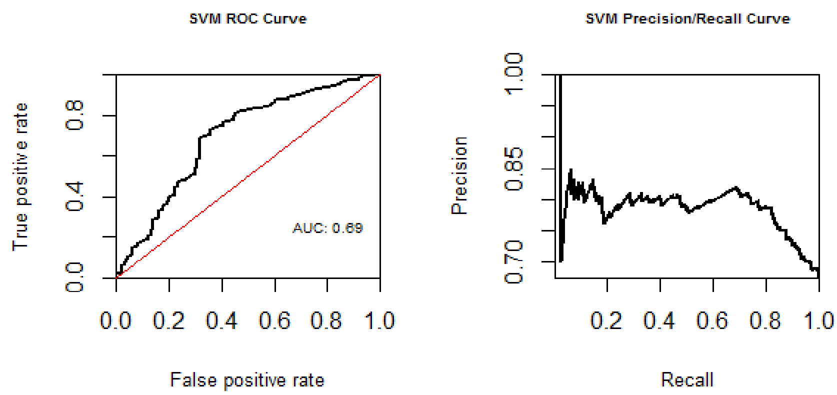


```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0  49  53
         1  81 217

               Accuracy : 0.665
                 95% CI : (0.6164, 0.7111)
    No Information Rate : 0.675
    P-Value [Acc > NIR] : 0.68620

                  Kappa : 0.1913
 Mcnemar's Test P-Value : 0.01968

            Sensitivity : 0.8037
            Specificity : 0.3769
         Pos Pred Value : 0.7282
         Neg Pred Value : 0.4804
             Prevalence : 0.6750
         Detection Rate : 0.5425
   Detection Prevalence : 0.7450
      Balanced Accuracy : 0.5903

       'Positive' Class : 1
```

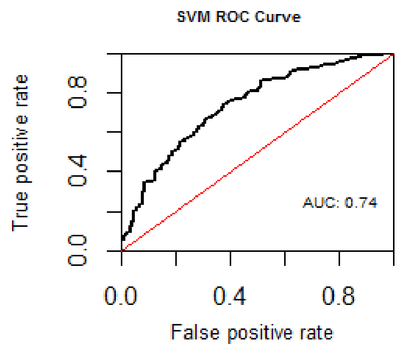
```
Parameter tuning of 'svm':
- sampling method: 10-fold cross validation
- best parameters:
 cost gamma
   10  0.25
- best performance: 0.22
```

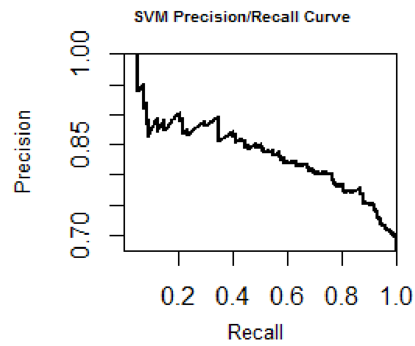


```
          Reference
Prediction   0   1
         0  53  38
         1  77 232

               Accuracy : 0.7125
                 95% CI : (0.6654, 0.7564)
    No Information Rate : 0.675
    P-Value [Acc > NIR] : 0.0596891

                  Kappa : 0.2895
 Mcnemar's Test P-Value : 0.0003948

            Sensitivity : 0.8593
            Specificity : 0.4077
```

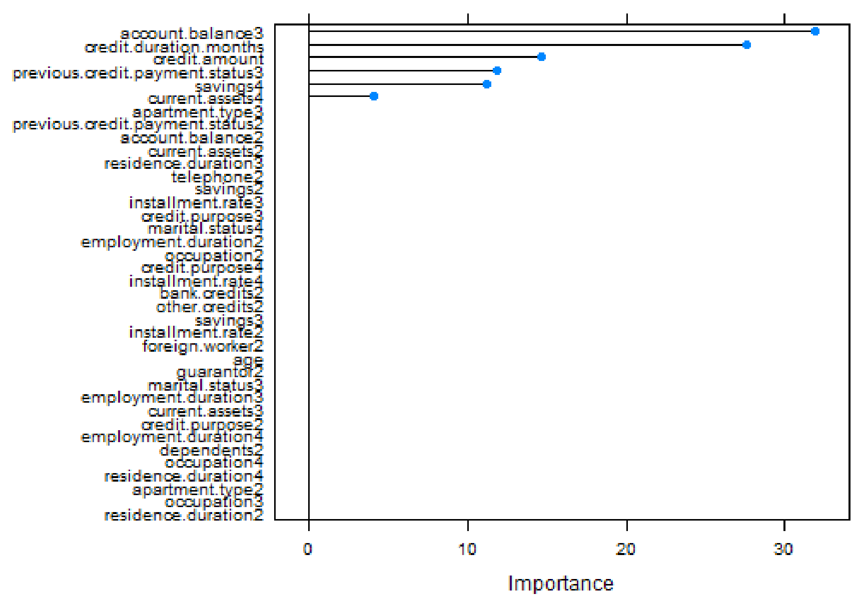



```
          Reference
Prediction  X0  X1
        X0  52  33
        X1  78 237

               Accuracy : 0.7225
                 95% CI : (0.6758, 0.7658)
    No Information Rate : 0.675
    P-Value [Acc > NIR] : 0.02302

                  Kappa : 0.3052
 Mcnemar's Test P-Value : 2.963e-05

            Sensitivity : 0.8778
            Specificity : 0.4000
```





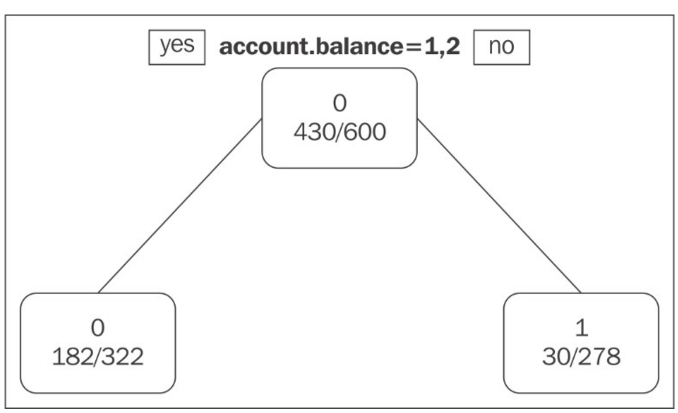

```
          Reference
Prediction   0   1
         0  23  22
         1 107 248

               Accuracy : 0.6775
                 95% CI : (0.6293, 0.7231)
    No Information Rate : 0.675
    P-Value [Acc > NIR] : 0.4812

                  Kappa : 0.1149
 Mcnemar's Test P-Value : 1.406e-13

            Sensitivity : 0.9185
            Specificity : 0.1769
```

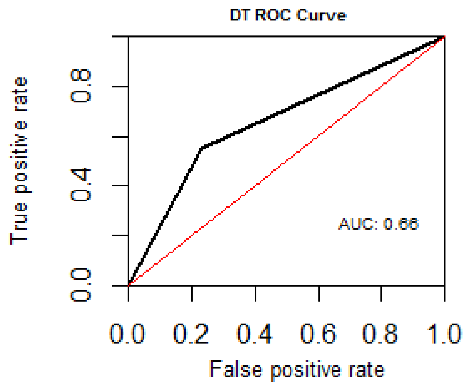


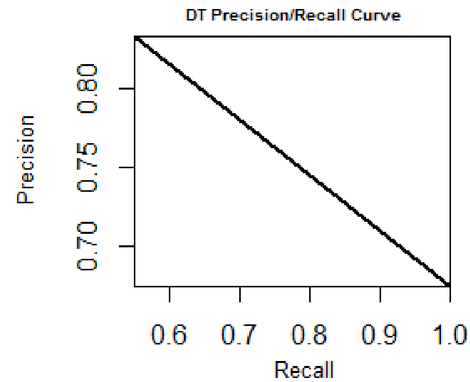
```
              Reference
Prediction    0   1
         0 100 121
         1  30 149

              Accuracy : 0.6225
                95% CI : (0.573, 0.6702)
   No Information Rate : 0.675
   P-Value [Acc > NIR] : 0.9884
                 Kappa : 0.2718
Mcnemar's Test P-Value : 2.405e-13
           Sensitivity : 0.5519
           Specificity : 0.7692
```

```
              Reference
Prediction    0   1
         0  23  22
         1 107 248

              Accuracy : 0.6775
                95% CI : (0.6293, 0.7231)
   No Information Rate : 0.675
   P-Value [Acc > NIR] : 0.4812
                 Kappa : 0.1149
Mcnemar's Test P-Value : 1.406e-13
           Sensitivity : 0.9185
           Specificity : 0.1769
```

```
n= 600

node), split, n, loss, yval, (yprob)
      * denotes terminal node

1) root 600 180.00000 0 (0.7000000 0.3000000)
  2) account.balance=1,2 322  76.18605 0 (0.8194936 0.1805064) *
  3) account.balance=3 278  74.11765 1 (0.4165513 0.5834487) *
```







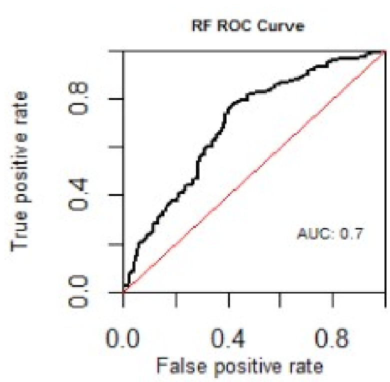

```
No. of variables tried at each split: 4
        OOB estimate of  error rate: 23.67%
Confusion matrix:
   0   1 class.error
0 61 109  0.64117647
1 33 397  0.07674419
```

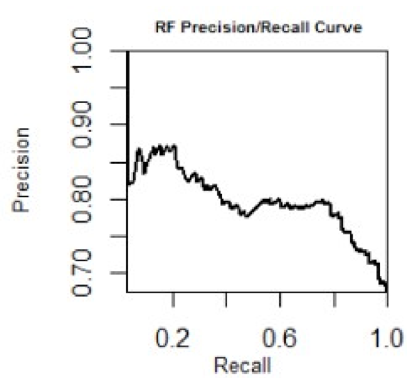
```
          Reference
Prediction   0   1
         0  47  24
         1  83 246
                                          
               Accuracy : 0.7325          
                 95% CI : (0.6863, 0.7753)
    No Information Rate : 0.675           
    P-Value [Acc > NIR] : 0.007434        
                  Kappa : 0.309           
Mcnemar's Test P-Value : 2.058e-08       
            Sensitivity : 0.9111          
            Specificity : 0.3615          
```

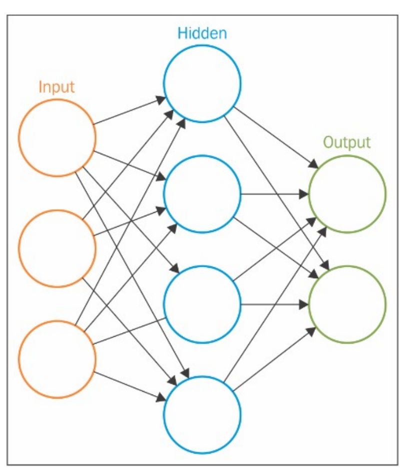
```
          Reference
Prediction   0   1
         0  55  42
         1  75 228
                                          
               Accuracy : 0.7075          
                 95% CI : (0.6602, 0.7517)
    No Information Rate : 0.675           
    P-Value [Acc > NIR] : 0.090176        
                  Kappa : 0.2864          
Mcnemar's Test P-Value : 0.003092        
            Sensitivity : 0.8444          
            Specificity : 0.4231          
```

```
Parameter tuning of 'randomForest':
- sampling method: 10-fold cross validation
- best parameters:
 nodesize mtry ntree
        5    3   500
- best performance: 0.215
```

```
          Reference
Prediction   0   1
         0  55  41
         1  75 229
               Accuracy : 0.71
                 95% CI : (0.6628, 0.754)
    No Information Rate : 0.675
    P-Value [Acc > NIR] : 0.073747
                  Kappa : 0.291
 Mcnemar's Test P-Value : 0.002184
            Sensitivity : 0.8481
            Specificity : 0.4231
```







$$\frac{TN}{FP+TN}$$

$$\frac{TP}{FN+TP}$$

$$\frac{TP}{FP+TP}$$

$$\frac{TN}{FN+TN}$$

$$\frac{FP}{FP+TN}$$

$$\frac{FN}{TP+FN}$$

$$\frac{TP+TN}{P+N}$$

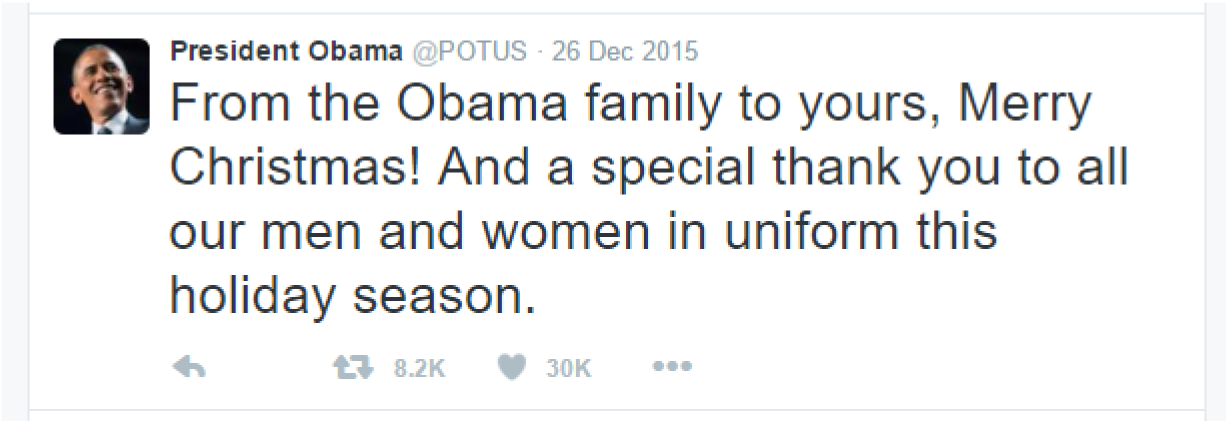
$$\frac{2TP}{2TP+FP+FN}$$

$$e_i$$

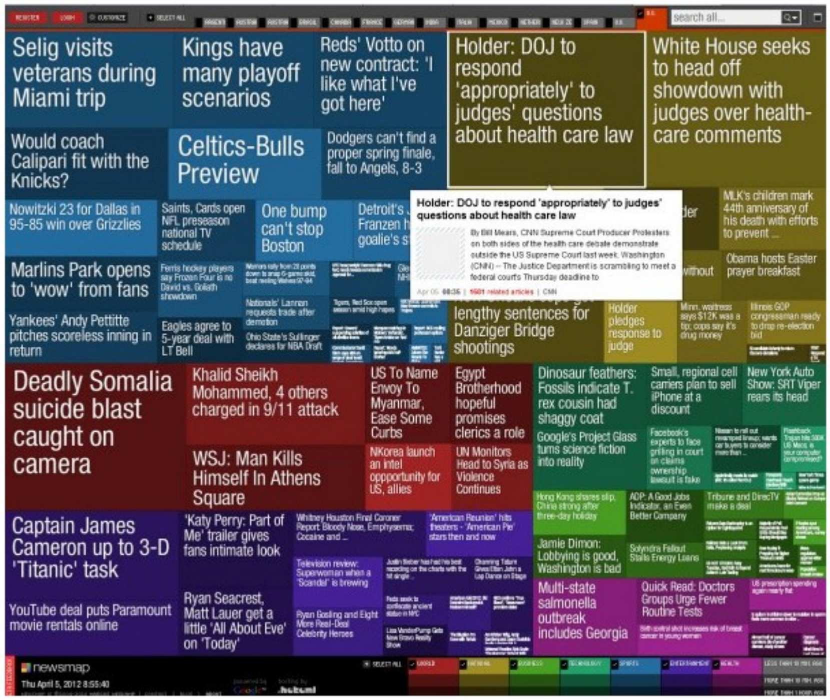
$$Znorm\left(e_i\right)=\frac{e_i-\overline{E}}{\sigma\left(E\right)}$$

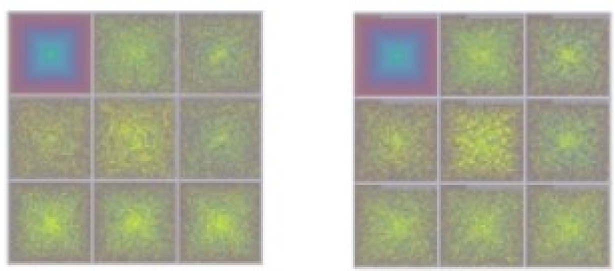
$$\overline{E}$$

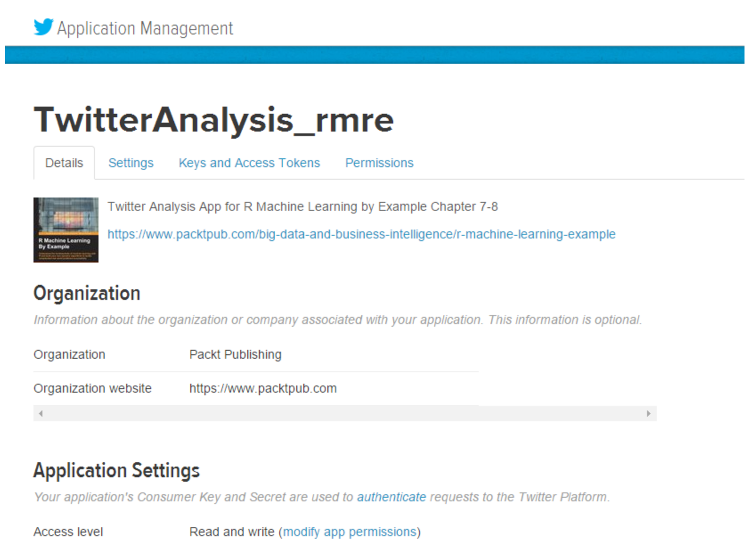
$$\sigma(E)$$


President Obama @POTUS · 26 Dec 2015
From the Obama family to yours, Merry Christmas! And a special thank you to all our men and women in uniform this holiday season.
8.2K
30K


lifecoach
get
learn
goals
slightly
mom
dailyquote
presidentâ€
call
credit
keeping
personaldevelopment
look
keep
help
motivation
become
trillionaire
check
inspire
networking
mosalingua
weareshape
learning
work
will
today
broaden
adopt
habit
horizons
make
hope
just
join
que
win
need
golf
gym


Selig visits veterans during Miami trip
Kings have many playoff scenarios
Reds' Votto on new contract: 'I like what I've got here'
Holder: DOJ to respond 'appropriately' to judges' questions about health care law
White House seeks to head off showdown with judges over health-care comments
Would coach Calipari fit with the Knicks?
Celtics-Bulls Preview
Dodgers can't find a proper spring finale, fall to Angels, 8-3
Nowitzki 23 for Dallas in 95-85 win over Grizzlies
Saints, Cards open NFL preseason national TV schedule
One bump can't stop Boston
MLK's children mark 44th anniversary of his death with efforts to prevent ...
Marlins Park opens to 'wow' from fans
Obama hosts Easter prayer breakfast
Yankees' Andy Pettitte pitches scoreless inning in return
Eagles agree to 5-year deal with LT Bell
lengthy sentences for Danziger Bridge shootings
Deadly Somalia suicide blast caught on camera
Khalid Sheikh Mohammed, 4 others charged in 9/11 attack
US To Name Envoy To Myanmar, Ease Some Curbs
Egypt Brotherhood hopeful promises clerics a role
Dinosaur feathers: Fossils indicate T. rex cousin had shaggy coat
Small, regional cell carriers plan to sell iPhone at a discount
New York Auto Show: SRT Viper rears its head
WSJ: Man Kills Himself In Athens Square
NKorea launch an intel opportunity for US, allies
UN Monitors Head to Syria as Violence Continues
Google's Project Glass turns science fiction into reality
Captain James Cameron up to 3-D 'Titanic' task
'Katy Perry: Part of Me' trailer gives fans intimate look
Jamie Dimon: Lobbying is good, Washington is bad
YouTube deal puts Paramount movie rentals online
Ryan Seacrest, Matt Lauer get a little 'All About Eve' on 'Today'
Multi-state salmonella outbreak includes Georgia
Quick Read: Doctors Groups Urge Fewer Routine Tests
newsmap
Thu April 5, 2012 8:55:40
search all...




Application Management
TwitterAnalysis_rmre
Details
Settings
Keys and Access Tokens
Permissions
Twitter Analysis App for R Machine Learning by Example Chapter 7-8
https://www.packtpub.com/big-data-and-business-intelligence/r-machine-learning-example
Organization
Information about the organization or company associated with your application. This information is optional.
Organization Packt Publishing
Organization website https://www.packtpub.com
Application Settings
Your application's Consumer Key and Secret are used to authenticate requests to the Twitter Platform.
Access level Read and write (modify app permissions)

TwitterAnalysis_rmre

Test OAuth

Details | Settings | Keys and Access Tokens | Permissions

Application Settings

Keep the "Consumer Secret" a secret. This key should never be human-readable in your application.

Consumer Key (API Key)	[redacted]
Consumer Secret (API Secret)	[redacted]
Access Level	Read and write (modify app permissions)
Owner	Rghv_Bali
Owner ID	48319844

Application Actions

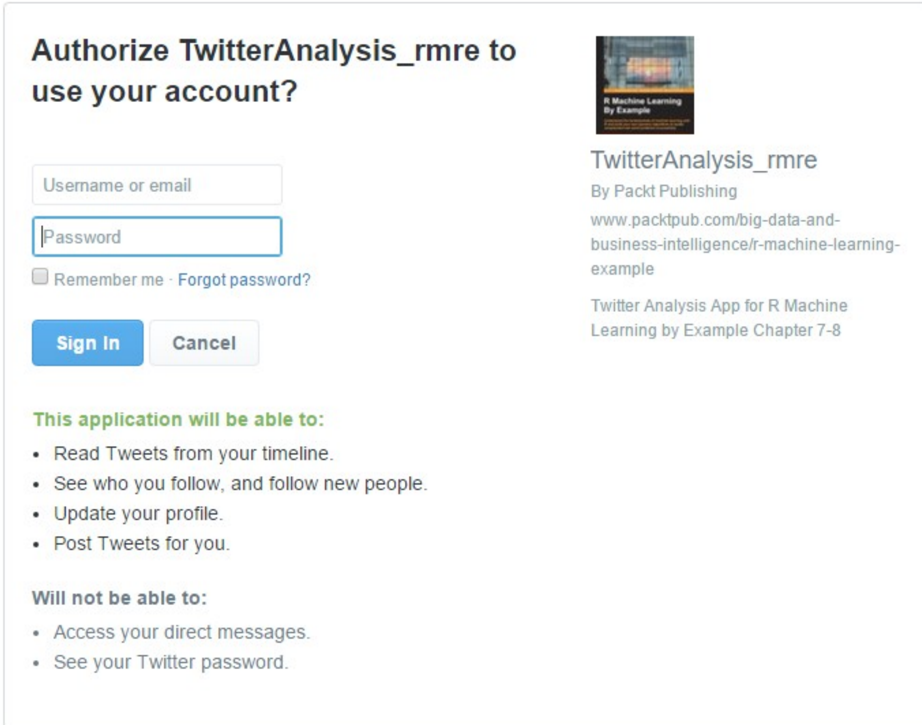
Regenerate Consumer Key and Secret | Change App Permissions

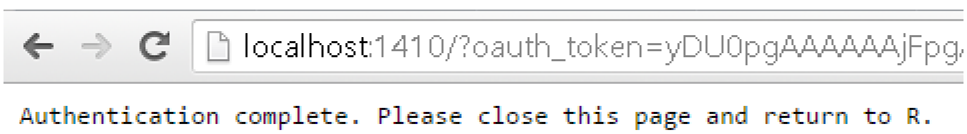
Your Access Token

This access token can be used to make API requests on your own account's behalf. Do not share your access token secret with anyone.

Access Token	[redacted]
Access Token Secret	[redacted]
Access Level	Read and write
Owner	Rghv_Bali
Owner ID	48319844

```
[1] "Using browser based authentication"
Use a local file to cache OAuth access credentials between R sessions?
1: Yes
2: No

Selection:
```






```
[[1]]
[1] "jack: \xed��\xed�\u0099@KillerMike Represents @BernieSanders in Post-Debate Spin Room /@p
itchfork  https://t.co/60sdEhZ18f"

[[2]]
[1] "jack: The @NBA goes all-in for MLK Day tribute with impact /@USATODAY  https://t.co/SF068gs
fuk"

[[3]]
[1] "jack: \"The time is always right to do what's right.\" https://t.co/V0fIVeLpY8"
```

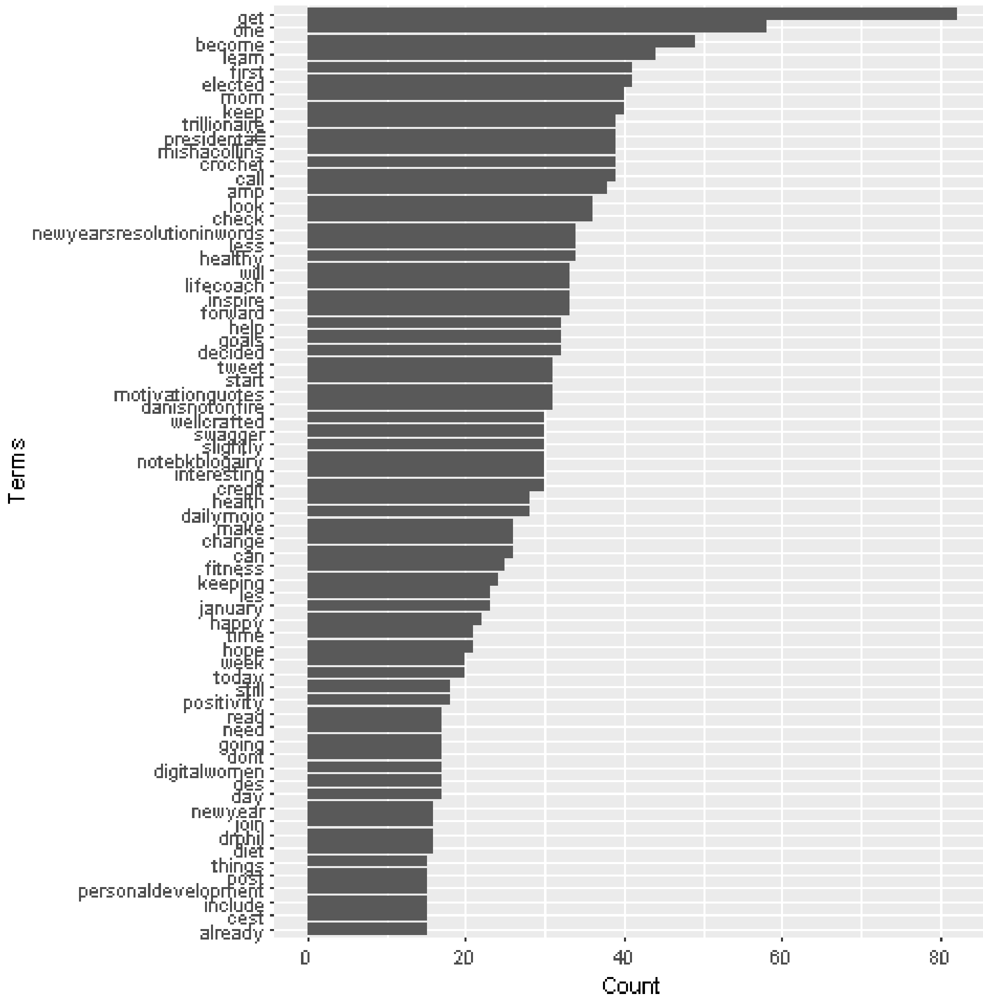
```
> # get tweet attributes
> tweets[[1]]$getClass()
Reference Class "status":

Class fields:

Name:          text     favorited favoriteCount     replyToSN       created     truncated    replyToSID            id
Class:    character       logical       numeric     character       POSIXct       logical     character     character

Name:     replyToUID  statusSource    screenName  retweetCount     isRetweet     retweeted     longitude      latitude
Class:    character     character     character       numeric       logical       logical     character     character

Name:          urls
Class:   data.frame

Class Methods:
     "setUrls", "getRetweets", "getRefClass", "getUrls", "setTruncated", "setText", "getReplyToSID", "getText", "export",
     "setCreated", "setFavoriteCount", "getCreated", "initialize", "callSuper", "getRetweeters", "initFields", "getClass",
     "setReplyToUID", "import", "setLatitude", "setIsRetweet", "getFavoriteCount", "getRetweetCount", "getIsRetweet", "setId",
     "setScreenName", "getLatitude", "getScreenName", "toDataFrame#twitterObj", "setRetweetCount", "setReplyToSID", "getId",
     "getReplyToUID", "setFavorited", "getRetweeted", "getFavorited", "toDataFrame", "setStatusSource", "setReplyToSN", "copy",
     "usingMethods", "setRetweeted", "field", ".objectParent", "getTruncated", "untrace", "trace", "setLongitude",
     "getLongitude", "getStatusSource", ".objectPackage", "getReplyToSN", "show"

Reference Superclasses:
     "twitterObj", "envRefClass"

> # get retweets count
> tweets[[1]]$retweetCount
[1] 21
> # get favourite count
> tweets[[1]]$favoriteCount
[1] 47
```

TD1:	I tweet all the time!
TD2:	I rarely tweet.

	TD1	TD2
tweet	1	1
all	1	0
time	1	0
rarely	0	1

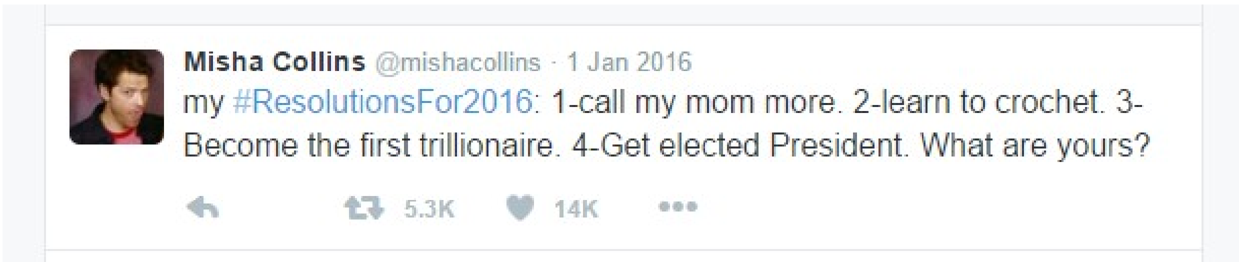
```
> which(apply(twtrTermDocMatrix,1,sum)>=30)
                       amp                      check                     credit
                        66                        297                        406
                 dailymojo                        get                    healthy
                       425                        710                        806
                   inspire                       keep                       less
                       901                        962                       1005
                 lifecoach                       look            motivationquotes
                      1014                       1042                       1182
newyearsresolutioninwords               notebkblogairy                        one
                      1229                       1246                       1281
                positivity                    swagger
                      1417                       1807
```

```
> (frequentTerms<-findFreqTerms(twtrTermDocMatrix,lowfreq = 10))
  [1] "adopt"                      "already"                    "amp"
  [4] "avec"                       "become"                     "book"
  [7] "broaden"                    "business"                   "call"
 [10] "can"                        "cest"                       "change"
 [13] "check"                      "credit"                     "crochet"
 [16] "daiâ€"                      "dailymojo"                  "dailyquote"
 [19] "danisnotonfire"             "day"                        "decided"
 [22] "des"                        "diet"                       "digitalwomen"
 [25] "dont"                       "elected"                    "end"
 [28] "faith"                      "femticdev"                  "find"
 [31] "first"                      "fitness"                    "forward"
 [34] "get"                        "give"                       "goal"
 [37] "goals"                      "going"                      "golf"
 [40] "gym"                        "habit"                      "happy"
 [43] "health"                     "healthy"                    "healthyliving"
 [46] "help"                       "hope"                       "horizons"
 [49] "include"                    "infographic"                "inspire"
 [52] "interesting"                "january"                    "join"
 [55] "just"                       "keep"                       "keeping"
 [58] "language"                   "learn"                      "learning"
 [61] "les"                        "less"                       "lifecoach"
 [64] "like"                       "look"                       "make"
 [67] "mishacollins"               "mom"                        "mosalingua"
 [70] "motivation"                 "motivationquotes"           "narechh"
 [73] "need"                       "networking"                 "newyear"
 [76] "newyearsresolutioninwords"  "noellbernard"               "notebkblogairy"
 [79] "nous"                       "one"                        "personaldevelopment"
 [82] "positivity"                 "post"                       "presidentâ€"
 [85] "que"                        "slightly"                   "start"
 [88] "started"                    "still"                      "success"
 [91] "swagger"                    "things"                     "time"
 [94] "tips"                       "today"                      "trillionaire"
 [97] "try"                        "tweet"                      "via"
[100] "vos"                        "way"                        "weareshapr"
[103] "week"                       "wellcrafted"                "will"
[106] "win"                        "work"
```

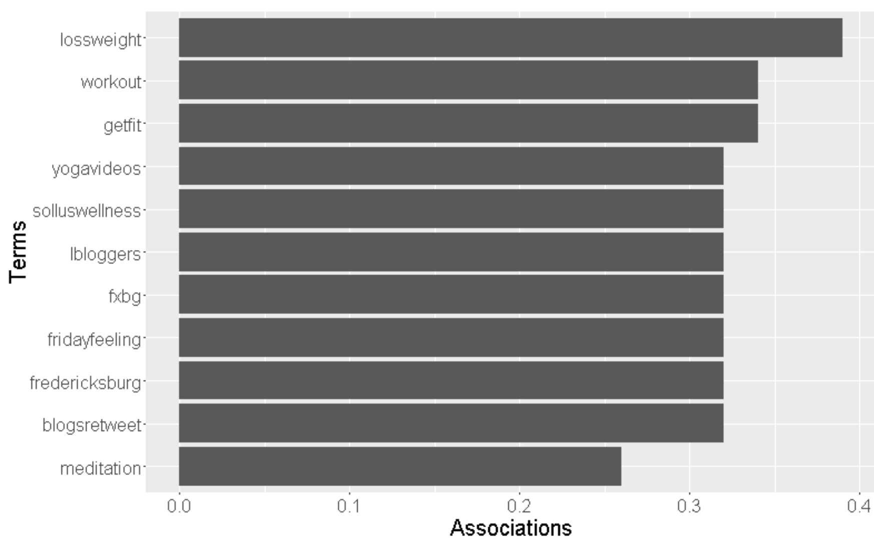

Terms
get
one
become
learn
first
elected
mom
keep
trillionaire
presidential
mishacollins
crochet
call
amp
look
check
newyearsresolutioninwords
less
healthy
will
lifecoach
inspire
forward
help
goals
decided
tweet
start
motivationquotes
danisnotonfire
wellcrafted
swagger
slightly
notebkblogairy
interesting
credit
health
dailymojo
make
change
can
fitness
keeping
les
january
happy
time
hope
week
today
still
positivity
read
need
going
dont
digitalwomen
des
day
newyear
join
drphil
diet
things
post
personaldevelopment
include
cest
already
0
20
40
60
80
Count

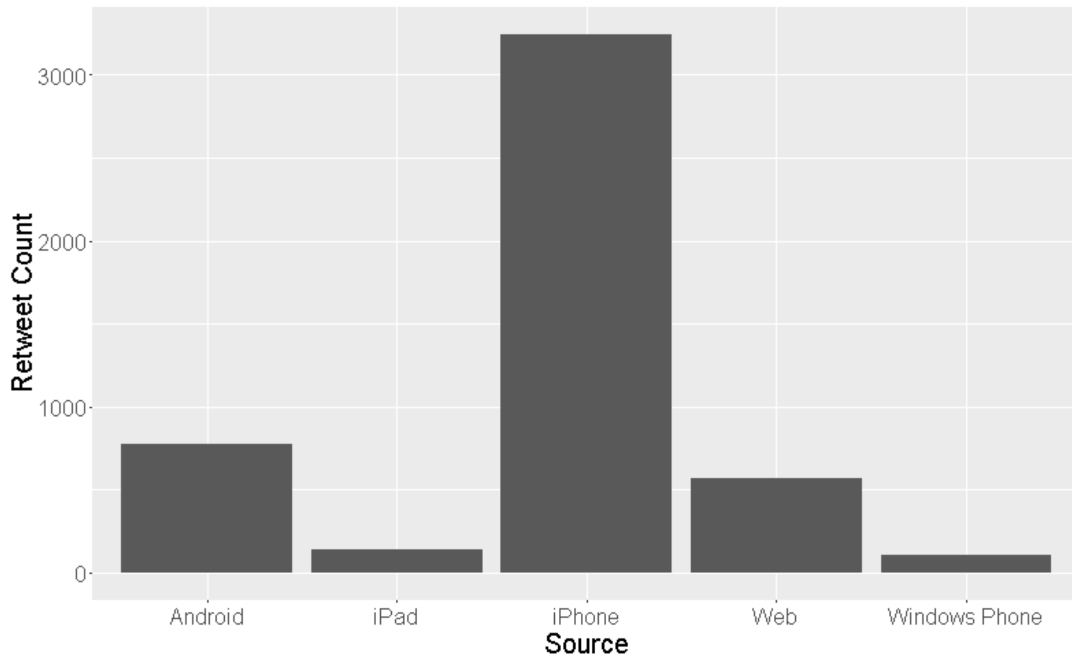

get
one
motivationquotes
inspire
lifecoach
dailymojo
amp
keep
notebkblogairy
become
trillionaire
wellcrafted
danisnotonfire
health
mom
look
check
less
call
credit
healthy
learn
slightly
swagger
tweet
forward
positivity
interesting
january
goals
decided
first
help
elected
change
happy
will
start
time
fitness
keeping
can
need
win
still
des
week
day
dont
make
les
try
give
habit
business
femticdev
find
join
end
golf
like
avec
eat faith
post
save
wonders
treat
mishboo
idea

```
> head(subset(trendingTweets.df$text, grepl("trillionaire",trendingTweets.df$text) ),n=1)
[1] "RT @mishacollins: my #ResolutionsFor2016: 1-call my mom more. 2-learn to crochet. 3-Become the first
trillionaire. 4-Get elected President.â€¦"
```

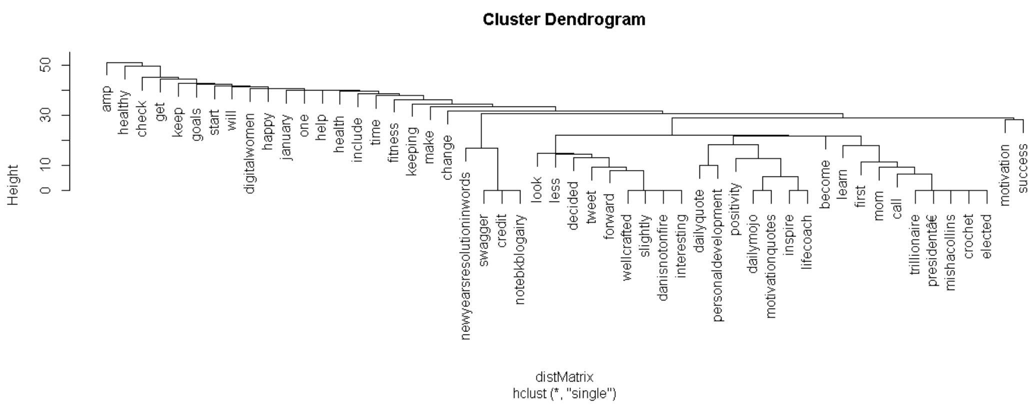



```
$fitness
    lossweight         getfit        workout    blogsretweet fredericksburg  fridayfeeling
          0.39           0.34           0.34            0.32           0.32           0.32
          fxbg      lbloggers solluswellness      yogavideos     meditation
          0.32           0.32           0.32            0.32           0.26
```





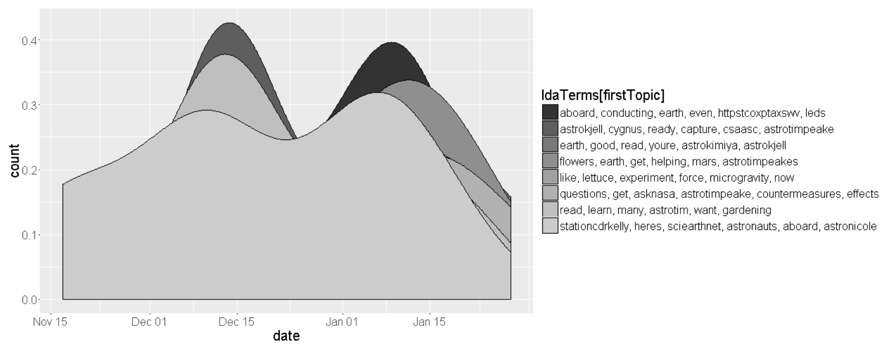
Retweet Count
3000
2000
1000
0
Android
iPad
iPhone
Web
Windows Phone
Source



Cluster Dendrogram
Height
50
30
10
0
amp
healthy
check
get
keep
goals
start
will
digitalwomen
happy
january
one
help
health
include
time
fitness
keeping
make
change
newyearsresolutioninwords
swagger
credit
notebkblogairy
look
less
decided
tweet
forward
wellcrafted
slightly
danisnotonfire
interesting
dailyquote
personaldevelopment
positivity
dailymojo
motivationquotes
inspire
lifecoach
become
learn
first
mom
call
trillionaire
presidentâ€
mishacollins
crochet
elected
motivation
success
distMatrix
hclust (*, "single")

susan garfinkel
@footnotesrising
Follow
"A topic is a recurring pattern of co-occurring words." now we know what we've been talking about today. #dhtopic

Topic 1
"like, lettuce, experiment, force, microgravity, now"
Topic 2
"flowers, earth, get, helping, mars, astrotimpeakes"
Topic 3
"questions, get, asknasa, astrotimpeake, countermeasures, effects"
Topic 4
"aboard, conducting, earth, even, httpstcoxptaxswv, leds"
Topic 5
"earth, good, read, youre, astrokimiya, astrokjell"
Topic 6
"astrokjell, cygnus, ready, capture, csaasc, astrotimpeake"
Topic 7
"stationcdrkelly, heres, sciearthnet, astronauts, aboard, astronicole"
Topic 8
"read, learn, many, astrotim, want, gardening"



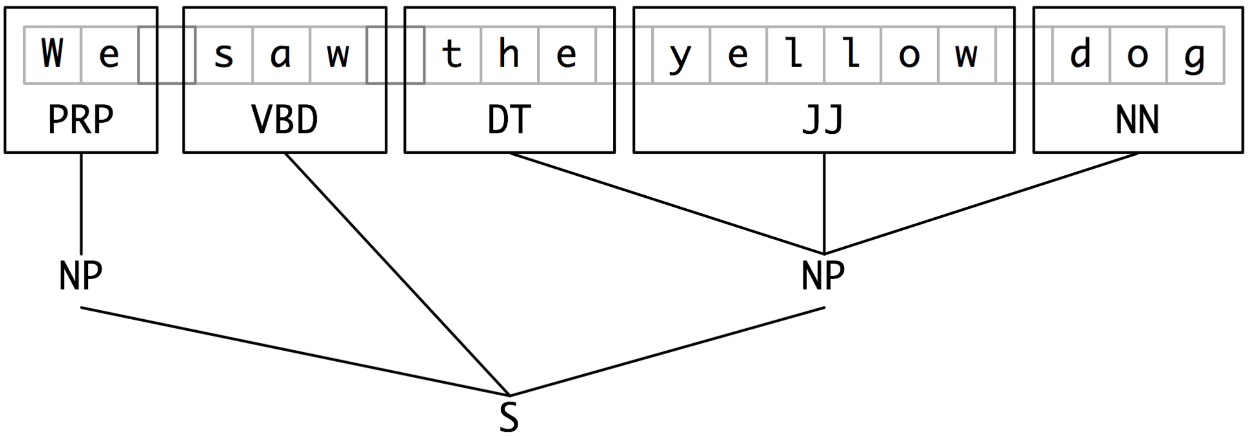
Chapter 8: Sentiment Analysis of Twitter Data

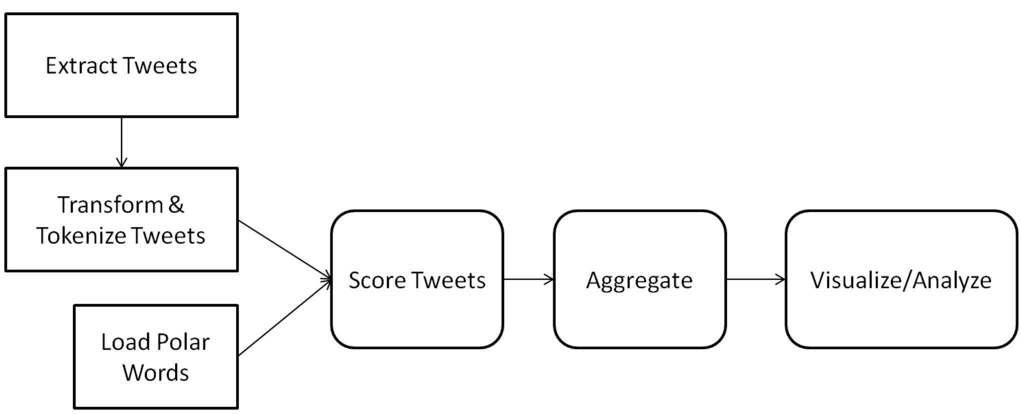
Document A

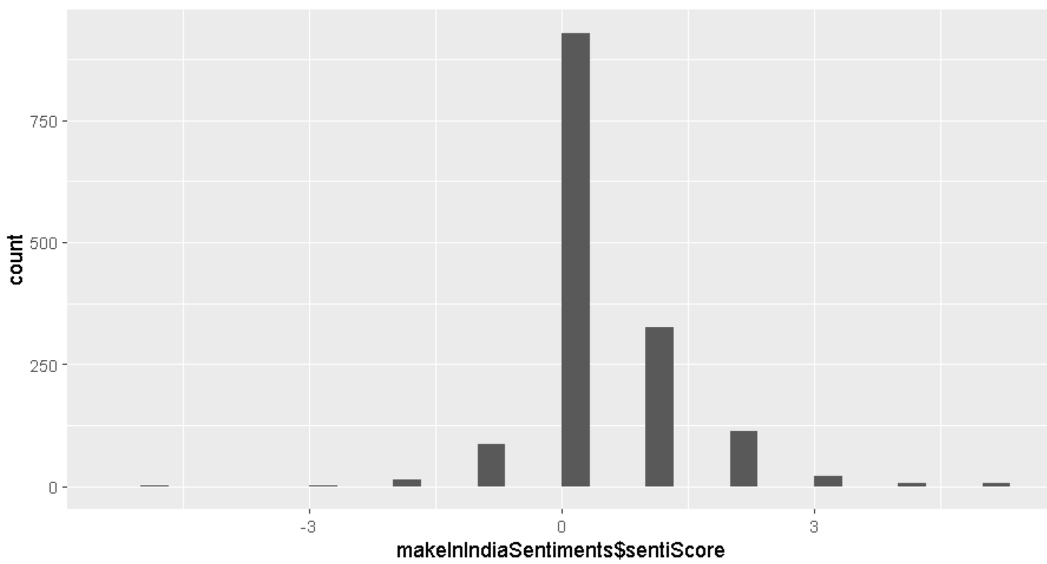
Term	Frequency
Twitter	3
internet	2
text	1
is	1

Document B

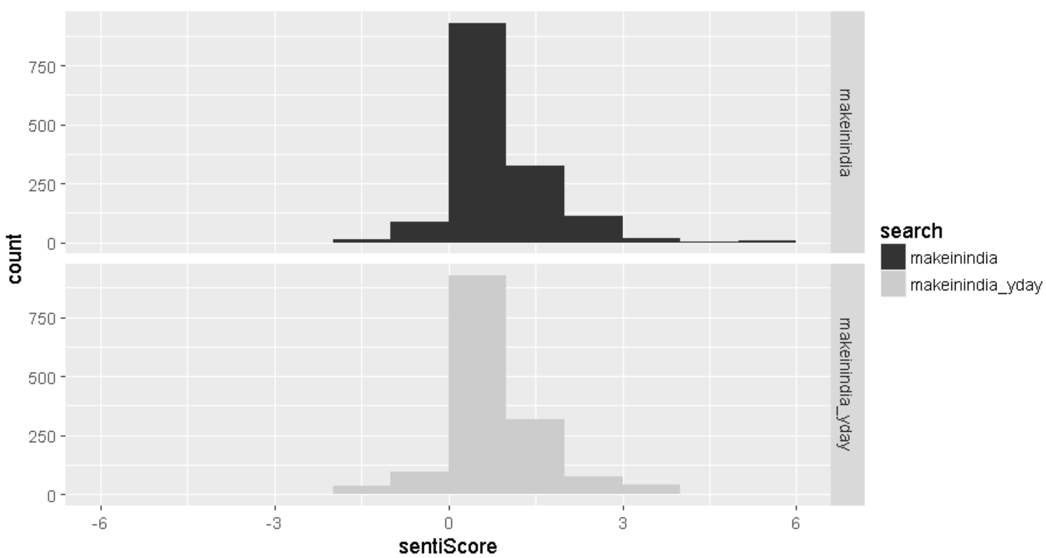
Term	Frequency
I	3
is	1
social	1
network	2



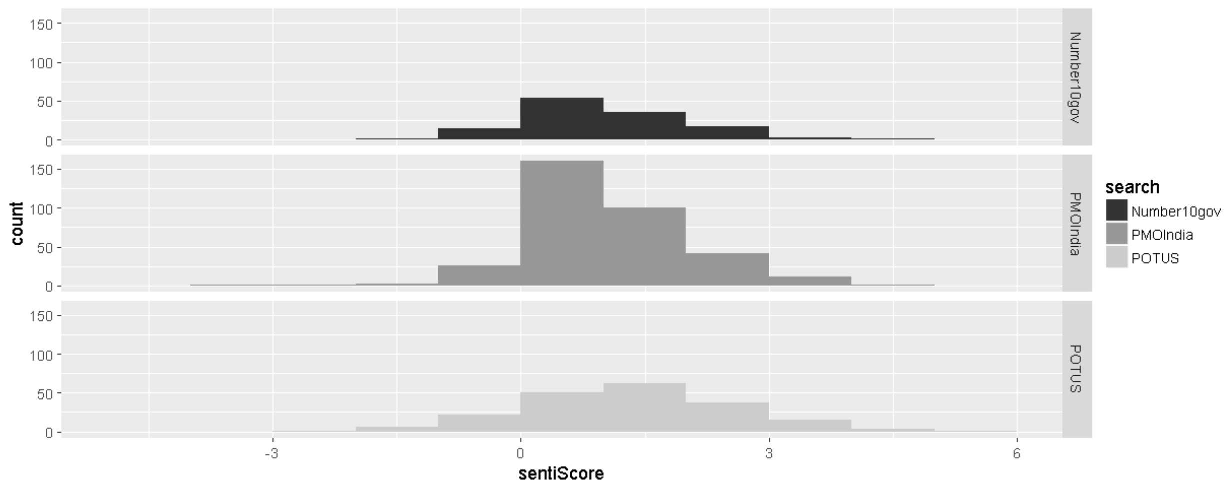




count
750
500
250
0
-3
0
3
makeInIndiaSentiments$sentiScore



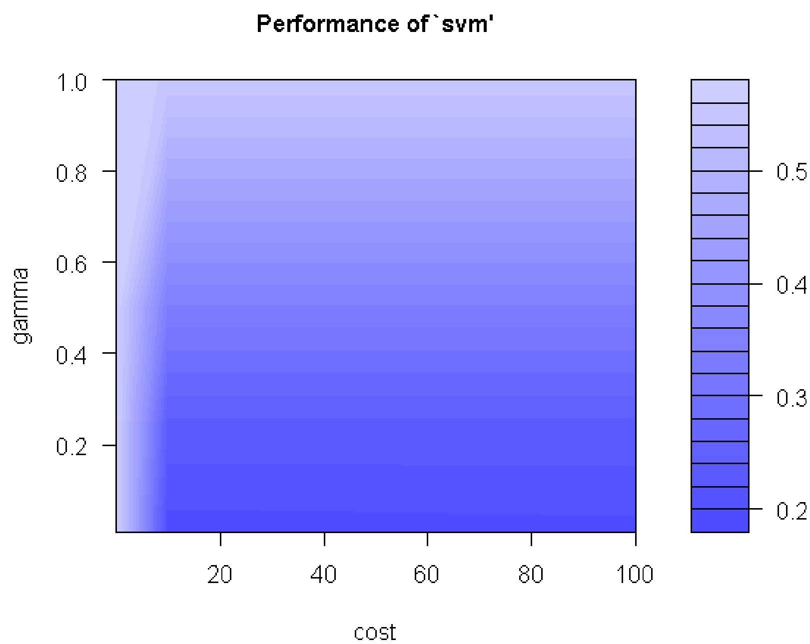
count
750
500
250
0
makeinindia
makeinindia_yday
search
-6
-3
0
3
6
sentiScore



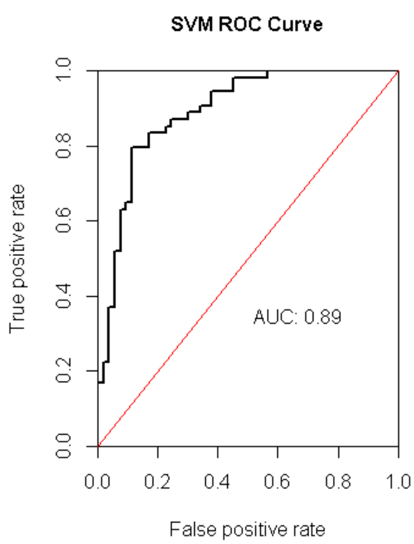
polarity	id	tweet_date	search qu	username	tweet text
4	12	Mon May 11 03:29:20 UTC 2009	obama	mandanicole	how can you not love Obama? he makes jokes about himself.
2	13	Mon May 11 03:32:42 UTC 2009	obama	jpeb	Check this video out -- President Obama at the White House Correspondents' Dinner http://bit.ly/IMXUM
0	14	Mon May 11 03:32:48 UTC 2009	obama	kylesellers	@Karoli I firmly believe that Obama/Pelosi have ZERO desire to be civil. It's a charade and a slogan, but they want to destroy

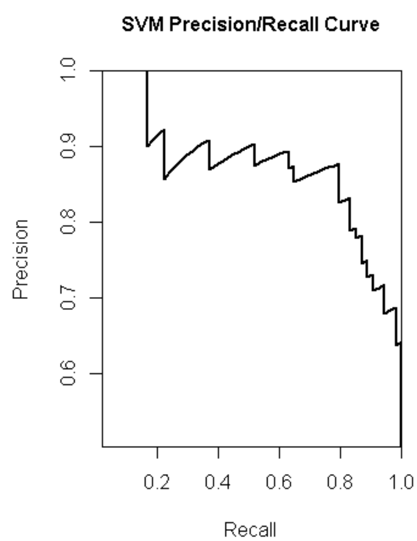
```
> prop.table(table(tweets[,1]))

 negative  positive
0.4930362 0.5069638
> prop.table(table(tweets[indexes,1]))

 negative  positive
0.4920635 0.5079365
> prop.table(table(tweets[-indexes,1]))

 negative  positive
0.4953271 0.5046729
```

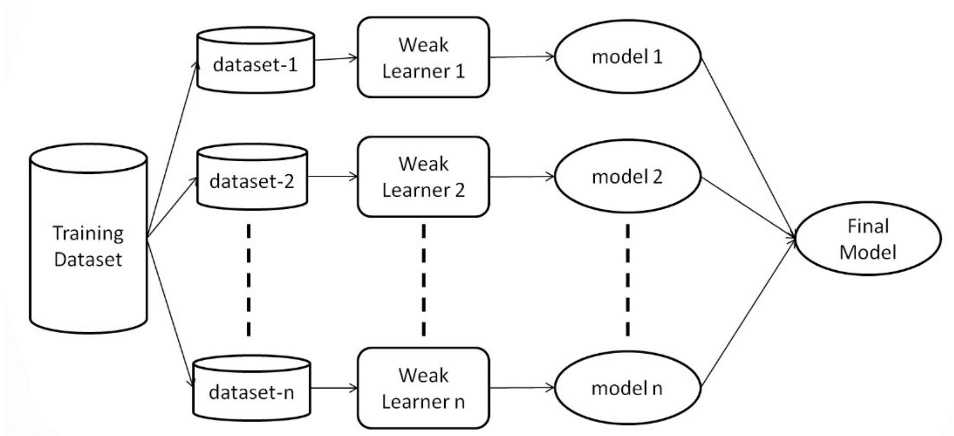
```
Confusion Matrix and Statistics

          Reference
Prediction negative positive
  negative        0        0
  positive       53       54

               Accuracy : 0.5047
                 95% CI : (0.4063, 0.6028)
    No Information Rate : 0.5047
    P-Value [Acc > NIR] : 0.5386

                  Kappa : 0
 Mcnemar's Test P-Value : 9.148e-13

            Sensitivity : 1.0000
            Specificity : 0.0000
         Pos Pred Value : 0.5047
         Neg Pred Value :    NaN
             Prevalence : 0.5047
         Detection Rate : 0.5047
   Detection Prevalence : 1.0000
      Balanced Accuracy : 0.5000

       'Positive' Class : positive
```



```
Confusion Matrix and Statistics

          Reference
Prediction negative positive
  negative       40        7
  positive       13       47

               Accuracy : 0.8131
                 95% CI : (0.7262, 0.8819)
    No Information Rate : 0.5047
    P-Value [Acc > NIR] : 3.553e-11

                  Kappa : 0.6257
 Mcnemar's Test P-Value : 0.2636

            Sensitivity : 0.8704
            Specificity : 0.7547
         Pos Pred Value : 0.7833
         Neg Pred Value : 0.8511
             Prevalence : 0.5047
         Detection Rate : 0.4393
   Detection Prevalence : 0.5607
      Balanced Accuracy : 0.8125

       'Positive' Class : positive
```







Given : $(x_1, y_1), \ldots, (x_m, y_m)$ where $x_i \in X, y_i \in Y = \{-1, +1\}$

Initialize $D_1(i) = 1/m$

For $t = 1, \ldots, T$:

- Train base learner using distribution D_t
- Get base classifier $h_t : X \to \mathbb{R}$
- Choose $\alpha_t \in \mathbb{R}$
- Update :

$$D_{t+1}(i) = \frac{D_t(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

where Z_t is a normalization factor (chosen so that D_{t+1} will be a distribution)

Output the final classifier :

$$H(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

```
Confusion Matrix and Statistics

          Reference
Prediction negative positive
  negative       40       10
  positive       13       44

               Accuracy : 0.785
                 95% CI : (0.6951, 0.8586)
    No Information Rate : 0.5047
    P-Value [Acc > NIR] : 2.128e-09

                  Kappa : 0.5698
 Mcnemar's Test P-Value : 0.6767

            Sensitivity : 0.8148
            Specificity : 0.7547
         Pos Pred Value : 0.7719
         Neg Pred Value : 0.8000
             Prevalence : 0.5047
         Detection Rate : 0.4112
   Detection Prevalence : 0.5327
      Balanced Accuracy : 0.7848

       'Positive' Class : positive
```

```
Fold 1 Out of Sample Accuracy = 1
Fold 2 Out of Sample Accuracy = 0.9344262
Fold 3 Out of Sample Accuracy = 1
Fold 4 Out of Sample Accuracy = 0.9310345
Fold 5 Out of Sample Accuracy = 1
Fold 6 Out of Sample Accuracy = 0.9591837
Fold 7 Out of Sample Accuracy = 1
Fold 8 Out of Sample Accuracy = 1
Fold 9 Out of Sample Accuracy = 1
Fold 10 Out of Sample Accuracy = 0.9649123
[[1]]
 [1] 1.0000000 0.9344262 1.0000000 0.9310345 1.0000000 0.9591837 1.0000000 1.0000000 1.0000000
[10] 0.9649123

$meanAccuracy
[1] 0.9789557
```


```
Fold 1 Out of Sample Accuracy = 1
Fold 2 Out of Sample Accuracy = 0.9344262
Fold 3 Out of Sample Accuracy = 1
Fold 4 Out of Sample Accuracy = 0.9310345
Fold 5 Out of Sample Accuracy = 1
Fold 6 Out of Sample Accuracy = 0.9591837
Fold 7 Out of Sample Accuracy = 1
Fold 8 Out of Sample Accuracy = 1
Fold 9 Out of Sample Accuracy = 1
Fold 10 Out of Sample Accuracy = 0.9649123
[[1]]
 [1] 1.0000000 0.9344262 1.0000000 0.9310345 1.0000000 0.9591837 1.0000000 1.0000000 1.0000000
[10] 0.9649123

$meanAccuracy
[1] 0.9789557
```


$$tfidf\left(t,d,D\right)=tf\left(t,d\right).idf\left(t,D\right)$$

$$
\begin{aligned}
tfifd\left(Twitter\right) &= tf\left(Twitter, Document\ A\right).idf\left(Twitter, Document\ Set\ D\right) \\
&= 3X0.3010 = 0.9030
\end{aligned}
$$