Ch. 06

JAMM

19/07/2021

Naive Bayes

To predict party affiliation. Looking for common voting paterns from previous elections. Train wether a congressperson was a D or a R. USARLO PARA CLASIFICAR SI UNA ACCION ES RENTABLE O NO\

```
## # A tibble: 435 x 17
##    Class V1    V2    V3    V4    V5    V6    V7    V8    V9    V10   V11   V12
##    <fct> <fct> <fct> <fct> <fct> <fct> <fct> <fct> <fct> <fct> <fct> <fct> <fct>
##  1 repu~ n     y     n     y     y     y     n     n     n     y     <NA>  y
##  2 repu~ n     y     n     y     y     y     n     n     n     n     n     y
##  3 demo~ <NA>  y     y     <NA>  y     y     n     n     n     n     y     n
##  4 demo~ n     y     y     n     <NA>  y     n     n     n     n     y     n
##  5 demo~ y     y     y     n     y     y     n     n     n     n     y     <NA>
##  6 demo~ n     y     y     n     y     y     n     n     n     n     n     n
##  7 demo~ n     y     n     y     y     y     n     n     n     n     n     n
##  8 repu~ n     y     n     y     y     y     n     n     n     n     n     n
##  9 repu~ n     y     n     y     y     y     n     n     n     n     n     y
## 10 demo~ y     y     y     n     n     n     y     y     y     n     n     n
## # ... with 425 more rows, and 4 more variables: V13 <fct>, V14 <fct>,
## #   V15 <fct>, V16 <fct>
```

The DB has `length(votesTib$Class)` by `length(votesTib[,1])`. Each of the columns is a factor. Each of them tell whether they voted, not voted or abstained.\ The number of missing values in the tibble is

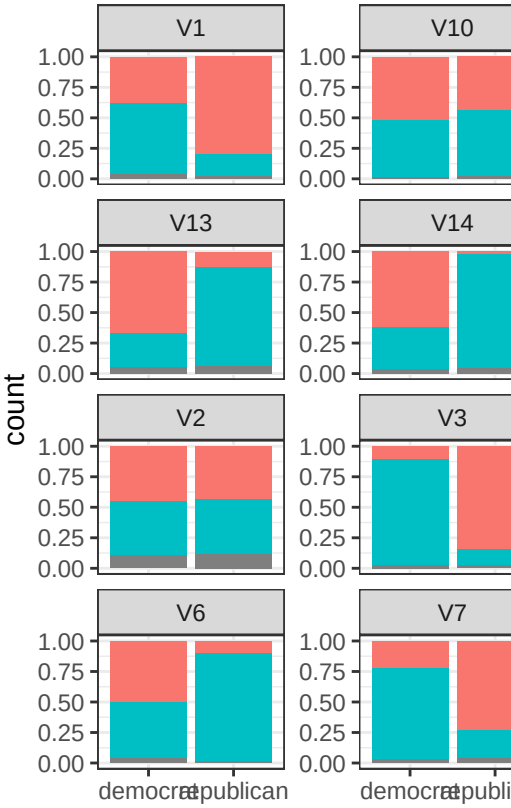
```
#First argument  of the  function  is the  name of the data. Second is the function we will apply to it
map_dbl(votesTib, ~sum(is.na(.)))
```

```
## Class    V1    V2    V3    V4    V5    V6    V7    V8    V9   V10   V11   V12
##     0    12    48    11    11    15    11    14    15    22     7    21    31
##   V13   V14   V15   V16
##    25    17    28   104
```

We can handle the NAs in 2 ways with Naive Bayes. One, ommiting them but using them to train the model. Two, ommiting it entirely.\ If we want to count the number of 'y'.

```
## Class    V1    V2    V3    V4    V5    V6    V7    V8    V9   V10   V11   V12
##     0   187   195   253   177   212   272   239   242   207   216   150   171
##   V13   V14   V15   V16
##   209   248   174   269
```

6.2.2 Plotting Data



We take into consideration that we are plotting categorical data against each other.

6.2.3 Training Model

Class variable -> classification target for makeClassifTask()\ The algorithm we'll use is "classif.naiveBayes".

```
## Warning in makeTask(type = type, data = data, weights = weights, blocking =
## blocking, : Provided data is not a pure data.frame but from class tbl_df, hence
## it will be converted.
```

We now use 10-fold CV. We repeated 50 times.\ We ask for the false and positive rates in the measures argument.

```
## Resampling: repeated cross-validation
## Measures:             mmce      acc       fpr       fnr
## [Resample] iter 1:    0.0697674 0.9302326 0.0625000 0.0740741
## [Resample] iter 2:    0.1395349 0.8604651 0.1250000 0.1481481
## [Resample] iter 3:    0.1363636 0.8636364 0.0588235 0.1851852
## [Resample] iter 4:    0.0681818 0.9318182 0.0588235 0.0740741
## [Resample] iter 5:    0.0697674 0.9302326 0.0000000 0.1153846
## [Resample] iter 6:    0.1162791 0.8837209 0.2352941 0.0384615
## [Resample] iter 7:    0.1590909 0.8409091 0.1764706 0.1481481
## [Resample] iter 8:    0.0681818 0.9318182 0.0588235 0.0740741
## [Resample] iter 9:    0.0232558 0.9767442 0.0000000 0.0384615
```

```
## [Resample] iter 10:   0.1363636 0.8636364 0.0588235 0.1851852
## [Resample] iter 11:   0.1818182 0.8181818 0.2352941 0.1481481
## [Resample] iter 12:   0.0930233 0.9069767 0.0000000 0.1538462
## [Resample] iter 13:   0.0930233 0.9069767 0.0000000 0.1538462
## [Resample] iter 14:   0.0681818 0.9318182 0.0588235 0.0740741
## [Resample] iter 15:   0.1395349 0.8604651 0.0625000 0.1851852
## [Resample] iter 16:   0.0681818 0.9318182 0.1176471 0.0370370
## [Resample] iter 17:   0.0454545 0.9545455 0.0000000 0.0740741
## [Resample] iter 18:   0.0476190 0.9523810 0.0000000 0.0769231
## [Resample] iter 19:   0.1363636 0.8636364 0.1764706 0.1111111
## [Resample] iter 20:   0.0909091 0.9090909 0.1176471 0.0740741
## [Resample] iter 21:   0.0681818 0.9318182 0.0588235 0.0740741
## [Resample] iter 22:   0.0697674 0.9302326 0.0625000 0.0740741
## [Resample] iter 23:   0.0697674 0.9302326 0.0000000 0.1153846
## [Resample] iter 24:   0.1860465 0.8139535 0.3125000 0.1111111
## [Resample] iter 25:   0.0454545 0.9545455 0.0000000 0.0740741
## [Resample] iter 26:   0.0681818 0.9318182 0.1176471 0.0370370
## [Resample] iter 27:   0.1363636 0.8636364 0.0588235 0.1851852
## [Resample] iter 28:   0.1136364 0.8863636 0.1176471 0.1111111
## [Resample] iter 29:   0.1162791 0.8837209 0.0588235 0.1538462
## [Resample] iter 30:   0.0930233 0.9069767 0.0588235 0.1153846
## [Resample] iter 31:   0.0909091 0.9090909 0.0588235 0.1111111
## [Resample] iter 32:   0.1136364 0.8863636 0.0588235 0.1481481
## [Resample] iter 33:   0.0465116 0.9534884 0.1176471 0.0000000
## [Resample] iter 34:   0.1136364 0.8863636 0.0000000 0.1851852
## [Resample] iter 35:   0.1860465 0.8139535 0.1875000 0.1851852
## [Resample] iter 36:   0.0681818 0.9318182 0.1176471 0.0370370
## [Resample] iter 37:   0.0697674 0.9302326 0.0625000 0.0740741
## [Resample] iter 38:   0.0930233 0.9069767 0.0000000 0.1538462
## [Resample] iter 39:   0.1136364 0.8863636 0.1764706 0.0740741
## [Resample] iter 40:   0.0697674 0.9302326 0.0000000 0.1153846
## [Resample] iter 41:   0.0952381 0.9047619 0.0625000 0.1153846
## [Resample] iter 42:   0.0930233 0.9069767 0.0588235 0.1153846
## [Resample] iter 43:   0.0681818 0.9318182 0.0000000 0.1111111
## [Resample] iter 44:   0.0454545 0.9545455 0.0588235 0.0370370
## [Resample] iter 45:   0.1162791 0.8837209 0.1176471 0.1153846
```

```
## [Resample] iter 46:   0.1627907 0.8372093 0.1250000 0.1851852
## [Resample] iter 47:   0.0681818 0.9318182 0.0588235 0.0740741
## [Resample] iter 48:   0.1363636 0.8636364 0.2941176 0.0370370
## [Resample] iter 49:   0.0909091 0.9090909 0.0588235 0.1111111
## [Resample] iter 50:   0.1136364 0.8863636 0.0000000 0.1851852
## [Resample] iter 51:   0.1136364 0.8863636 0.2352941 0.0370370
## [Resample] iter 52:   0.0909091 0.9090909 0.1176471 0.0740741
## [Resample] iter 53:   0.0909091 0.9090909 0.1176471 0.0740741
## [Resample] iter 54:   0.1395349 0.8604651 0.0588235 0.1923077
## [Resample] iter 55:   0.1136364 0.8863636 0.0588235 0.1481481
## [Resample] iter 56:   0.0697674 0.9302326 0.0588235 0.0769231
## [Resample] iter 57:   0.1395349 0.8604651 0.1250000 0.1481481
## [Resample] iter 58:   0.1904762 0.8095238 0.0000000 0.3076923
## [Resample] iter 59:   0.0227273 0.9772727 0.0000000 0.0370370
## [Resample] iter 60:   0.0227273 0.9772727 0.0000000 0.0370370
## [Resample] iter 61:   0.0681818 0.9318182 0.0588235 0.0740741
## [Resample] iter 62:   0.0930233 0.9069767 0.1764706 0.0384615
## [Resample] iter 63:   0.1860465 0.8139535 0.1176471 0.2307692
## [Resample] iter 64:   0.0909091 0.9090909 0.1176471 0.0740741
## [Resample] iter 65:   0.1590909 0.8409091 0.0588235 0.2222222
## [Resample] iter 66:   0.0681818 0.9318182 0.0000000 0.1111111
## [Resample] iter 67:   0.0909091 0.9090909 0.0588235 0.1111111
## [Resample] iter 68:   0.1190476 0.8809524 0.1250000 0.1153846
## [Resample] iter 69:   0.0454545 0.9545455 0.0000000 0.0740741
## [Resample] iter 70:   0.0697674 0.9302326 0.1250000 0.0370370
## [Resample] iter 71:   0.1590909 0.8409091 0.1764706 0.1481481
## [Resample] iter 72:   0.0697674 0.9302326 0.0588235 0.0769231
## [Resample] iter 73:   0.1363636 0.8636364 0.1176471 0.1481481
## [Resample] iter 74:   0.0000000 1.0000000 0.0000000 0.0000000
## [Resample] iter 75:   0.0952381 0.9047619 0.0000000 0.1538462
## [Resample] iter 76:   0.0681818 0.9318182 0.0588235 0.0740741
## [Resample] iter 77:   0.0697674 0.9302326 0.0000000 0.1153846
## [Resample] iter 78:   0.1590909 0.8409091 0.1176471 0.1851852
## [Resample] iter 79:   0.1363636 0.8636364 0.1764706 0.1111111
## [Resample] iter 80:   0.0909091 0.9090909 0.1176471 0.0740741
## [Resample] iter 81:   0.1363636 0.8636364 0.1176471 0.1481481
```

```
## [Resample] iter 82:   0.0681818 0.9318182 0.1176471 0.0370370
## [Resample] iter 83:   0.0952381 0.9047619 0.0625000 0.1153846
## [Resample] iter 84:   0.0909091 0.9090909 0.0588235 0.1111111
## [Resample] iter 85:   0.1136364 0.8863636 0.0588235 0.1481481
## [Resample] iter 86:   0.0930233 0.9069767 0.1176471 0.0769231
## [Resample] iter 87:   0.1162791 0.8837209 0.1176471 0.1153846
## [Resample] iter 88:   0.1136364 0.8863636 0.1176471 0.1111111
## [Resample] iter 89:   0.0681818 0.9318182 0.0000000 0.1111111
## [Resample] iter 90:   0.0930233 0.9069767 0.0625000 0.1111111
## [Resample] iter 91:   0.0930233 0.9069767 0.0000000 0.1538462
## [Resample] iter 92:   0.0909091 0.9090909 0.1764706 0.0370370
## [Resample] iter 93:   0.1590909 0.8409091 0.1176471 0.1851852
## [Resample] iter 94:   0.0930233 0.9069767 0.1764706 0.0384615
## [Resample] iter 95:   0.0909091 0.9090909 0.1176471 0.0740741
## [Resample] iter 96:   0.1162791 0.8837209 0.0000000 0.1851852
## [Resample] iter 97:   0.0454545 0.9545455 0.0000000 0.0740741
## [Resample] iter 98:   0.1395349 0.8604651 0.0625000 0.1851852
## [Resample] iter 99:   0.0465116 0.9534884 0.0000000 0.0769231
## [Resample] iter 100:  0.1136364 0.8863636 0.1764706 0.0740741
## [Resample] iter 101:  0.0465116 0.9534884 0.0000000 0.0740741
## [Resample] iter 102:  0.0697674 0.9302326 0.1764706 0.0000000
## [Resample] iter 103:  0.1162791 0.8837209 0.0000000 0.1923077
## [Resample] iter 104:  0.1590909 0.8409091 0.1176471 0.1851852
## [Resample] iter 105:  0.1363636 0.8636364 0.1176471 0.1481481
## [Resample] iter 106:  0.0909091 0.9090909 0.0588235 0.1111111
## [Resample] iter 107:  0.0697674 0.9302326 0.0625000 0.0740741
## [Resample] iter 108:  0.1136364 0.8863636 0.1764706 0.0740741
## [Resample] iter 109:  0.0930233 0.9069767 0.1176471 0.0769231
## [Resample] iter 110:  0.0909091 0.9090909 0.0000000 0.1481481
## [Resample] iter 111:  0.0697674 0.9302326 0.0588235 0.0769231
## [Resample] iter 112:  0.0697674 0.9302326 0.0625000 0.0740741
## [Resample] iter 113:  0.0681818 0.9318182 0.0588235 0.0740741
## [Resample] iter 114:  0.0465116 0.9534884 0.0000000 0.0740741
## [Resample] iter 115:  0.0909091 0.9090909 0.0000000 0.1481481
## [Resample] iter 116:  0.1590909 0.8409091 0.2352941 0.1111111
## [Resample] iter 117:  0.1860465 0.8139535 0.1764706 0.1923077
```

```
## [Resample] iter 118:  0.1136364 0.8863636 0.0588235 0.1481481
## [Resample] iter 119:  0.1136364 0.8863636 0.1176471 0.1111111
## [Resample] iter 120:  0.0697674 0.9302326 0.0588235 0.0769231
## [Resample] iter 121:  0.1395349 0.8604651 0.1764706 0.1153846
## [Resample] iter 122:  0.1190476 0.8809524 0.1250000 0.1153846
## [Resample] iter 123:  0.0681818 0.9318182 0.1176471 0.0370370
## [Resample] iter 124:  0.0909091 0.9090909 0.1176471 0.0740741
## [Resample] iter 125:  0.0232558 0.9767442 0.0000000 0.0384615
## [Resample] iter 126:  0.0909091 0.9090909 0.0000000 0.1481481
## [Resample] iter 127:  0.0681818 0.9318182 0.0588235 0.0740741
## [Resample] iter 128:  0.0909091 0.9090909 0.0588235 0.1111111
## [Resample] iter 129:  0.0909091 0.9090909 0.1176471 0.0740741
## [Resample] iter 130:  0.2093023 0.7906977 0.0625000 0.2962963
## [Resample] iter 131:  0.0909091 0.9090909 0.0588235 0.1111111
## [Resample] iter 132:  0.0681818 0.9318182 0.0000000 0.1111111
## [Resample] iter 133:  0.0697674 0.9302326 0.0000000 0.1111111
## [Resample] iter 134:  0.0930233 0.9069767 0.0588235 0.1153846
## [Resample] iter 135:  0.0465116 0.9534884 0.0000000 0.0769231
## [Resample] iter 136:  0.1627907 0.8372093 0.2352941 0.1153846
## [Resample] iter 137:  0.0909091 0.9090909 0.0588235 0.1111111
## [Resample] iter 138:  0.1136364 0.8863636 0.2352941 0.0370370
## [Resample] iter 139:  0.0909091 0.9090909 0.0000000 0.1481481
## [Resample] iter 140:  0.1395349 0.8604651 0.1250000 0.1481481
## [Resample] iter 141:  0.0930233 0.9069767 0.0000000 0.1481481
## [Resample] iter 142:  0.1860465 0.8139535 0.2352941 0.1538462
## [Resample] iter 143:  0.0697674 0.9302326 0.0588235 0.0769231
## [Resample] iter 144:  0.0681818 0.9318182 0.1176471 0.0370370
## [Resample] iter 145:  0.0681818 0.9318182 0.0588235 0.0740741
## [Resample] iter 146:  0.0465116 0.9534884 0.0000000 0.0740741
## [Resample] iter 147:  0.1136364 0.8863636 0.0588235 0.1481481
## [Resample] iter 148:  0.1162791 0.8837209 0.0000000 0.1923077
## [Resample] iter 149:  0.1363636 0.8636364 0.1764706 0.1111111
## [Resample] iter 150:  0.0681818 0.9318182 0.0588235 0.0740741
## [Resample] iter 151:  0.1395349 0.8604651 0.0588235 0.1923077
## [Resample] iter 152:  0.1395349 0.8604651 0.1176471 0.1538462
## [Resample] iter 153:  0.0681818 0.9318182 0.1764706 0.0000000
```

```
## [Resample] iter 154:  0.1136364 0.8863636 0.1764706 0.0740741
## [Resample] iter 155:  0.0930233 0.9069767 0.1176471 0.0769231
## [Resample] iter 156:  0.0465116 0.9534884 0.0625000 0.0370370
## [Resample] iter 157:  0.0930233 0.9069767 0.0000000 0.1481481
## [Resample] iter 158:  0.1136364 0.8863636 0.0588235 0.1481481
## [Resample] iter 159:  0.0681818 0.9318182 0.0588235 0.0740741
## [Resample] iter 160:  0.1136364 0.8863636 0.0000000 0.1851852
## [Resample] iter 161:  0.0909091 0.9090909 0.1764706 0.0370370
## [Resample] iter 162:  0.1136364 0.8863636 0.1176471 0.1111111
## [Resample] iter 163:  0.1363636 0.8636364 0.0588235 0.1851852
## [Resample] iter 164:  0.0681818 0.9318182 0.1176471 0.0370370
## [Resample] iter 165:  0.0681818 0.9318182 0.0000000 0.1111111
## [Resample] iter 166:  0.0714286 0.9285714 0.1250000 0.0384615
## [Resample] iter 167:  0.1136364 0.8863636 0.1176471 0.1111111
## [Resample] iter 168:  0.0697674 0.9302326 0.0588235 0.0769231
## [Resample] iter 169:  0.1860465 0.8139535 0.0000000 0.2962963
## [Resample] iter 170:  0.0697674 0.9302326 0.0588235 0.0769231
## [Resample] iter 171:  0.1363636 0.8636364 0.0588235 0.1851852
## [Resample] iter 172:  0.0930233 0.9069767 0.0588235 0.1153846
## [Resample] iter 173:  0.0454545 0.9545455 0.0000000 0.0740741
## [Resample] iter 174:  0.0465116 0.9534884 0.0000000 0.0740741
## [Resample] iter 175:  0.1136364 0.8863636 0.0000000 0.1851852
## [Resample] iter 176:  0.1363636 0.8636364 0.1764706 0.1111111
## [Resample] iter 177:  0.0930233 0.9069767 0.0588235 0.1153846
## [Resample] iter 178:  0.1136364 0.8863636 0.1764706 0.0740741
## [Resample] iter 179:  0.1162791 0.8837209 0.1176471 0.1153846
## [Resample] iter 180:  0.0930233 0.9069767 0.1875000 0.0370370
## [Resample] iter 181:  0.1904762 0.8095238 0.1250000 0.2307692
## [Resample] iter 182:  0.0697674 0.9302326 0.0625000 0.0740741
## [Resample] iter 183:  0.1136364 0.8863636 0.2352941 0.0370370
## [Resample] iter 184:  0.1162791 0.8837209 0.0588235 0.1538462
## [Resample] iter 185:  0.0909091 0.9090909 0.0000000 0.1481481
## [Resample] iter 186:  0.1136364 0.8863636 0.2352941 0.0370370
## [Resample] iter 187:  0.1363636 0.8636364 0.0000000 0.2222222
## [Resample] iter 188:  0.0465116 0.9534884 0.0588235 0.0384615
## [Resample] iter 189:  0.0681818 0.9318182 0.0588235 0.0740741
```

```
## [Resample] iter 190:  0.0681818 0.9318182 0.1176471 0.0370370
## [Resample] iter 191:  0.1363636 0.8636364 0.1764706 0.1111111
## [Resample] iter 192:  0.1363636 0.8636364 0.1176471 0.1481481
## [Resample] iter 193:  0.1162791 0.8837209 0.0000000 0.1851852
## [Resample] iter 194:  0.0697674 0.9302326 0.0588235 0.0769231
## [Resample] iter 195:  0.1428571 0.8571429 0.1250000 0.1538462
## [Resample] iter 196:  0.0681818 0.9318182 0.0000000 0.1111111
## [Resample] iter 197:  0.0681818 0.9318182 0.1764706 0.0000000
## [Resample] iter 198:  0.1136364 0.8863636 0.0588235 0.1481481
## [Resample] iter 199:  0.0930233 0.9069767 0.1176471 0.0769231
## [Resample] iter 200:  0.0681818 0.9318182 0.0588235 0.0740741
## [Resample] iter 201:  0.0909091 0.9090909 0.1764706 0.0370370
## [Resample] iter 202:  0.1627907 0.8372093 0.0625000 0.2222222
## [Resample] iter 203:  0.0909091 0.9090909 0.1764706 0.0370370
## [Resample] iter 204:  0.0454545 0.9545455 0.0000000 0.0740741
## [Resample] iter 205:  0.0681818 0.9318182 0.0000000 0.1111111
## [Resample] iter 206:  0.1860465 0.8139535 0.0588235 0.2692308
## [Resample] iter 207:  0.0000000 1.0000000 0.0000000 0.0000000
## [Resample] iter 208:  0.1904762 0.8095238 0.1875000 0.1923077
## [Resample] iter 209:  0.0930233 0.9069767 0.0588235 0.1153846
## [Resample] iter 210:  0.0454545 0.9545455 0.0588235 0.0370370
## [Resample] iter 211:  0.1666667 0.8333333 0.1250000 0.1923077
## [Resample] iter 212:  0.0909091 0.9090909 0.1176471 0.0740741
## [Resample] iter 213:  0.0232558 0.9767442 0.0000000 0.0384615
## [Resample] iter 214:  0.0909091 0.9090909 0.0588235 0.1111111
## [Resample] iter 215:  0.0909091 0.9090909 0.0000000 0.1481481
## [Resample] iter 216:  0.0681818 0.9318182 0.0588235 0.0740741
## [Resample] iter 217:  0.0697674 0.9302326 0.0588235 0.0769231
## [Resample] iter 218:  0.0930233 0.9069767 0.1250000 0.0740741
## [Resample] iter 219:  0.2045455 0.7954545 0.2352941 0.1851852
## [Resample] iter 220:  0.0909091 0.9090909 0.0588235 0.1111111
## [Resample] iter 221:  0.1162791 0.8837209 0.0625000 0.1481481
## [Resample] iter 222:  0.0465116 0.9534884 0.0588235 0.0384615
## [Resample] iter 223:  0.1136364 0.8863636 0.0588235 0.1481481
## [Resample] iter 224:  0.1666667 0.8333333 0.1875000 0.1538462
## [Resample] iter 225:  0.1136364 0.8863636 0.1764706 0.0740741
```

```
## [Resample] iter 226:  0.0681818 0.9318182 0.0588235 0.0740741
## [Resample] iter 227:  0.0909091 0.9090909 0.0588235 0.1111111
## [Resample] iter 228:  0.0930233 0.9069767 0.0000000 0.1538462
## [Resample] iter 229:  0.1363636 0.8636364 0.1176471 0.1481481
## [Resample] iter 230:  0.0454545 0.9545455 0.0588235 0.0370370
## [Resample] iter 231:  0.1136364 0.8863636 0.1764706 0.0740741
## [Resample] iter 232:  0.0476190 0.9523810 0.0625000 0.0384615
## [Resample] iter 233:  0.0930233 0.9069767 0.1176471 0.0769231
## [Resample] iter 234:  0.0697674 0.9302326 0.0625000 0.0740741
## [Resample] iter 235:  0.1162791 0.8837209 0.0588235 0.1538462
## [Resample] iter 236:  0.0681818 0.9318182 0.0000000 0.1111111
## [Resample] iter 237:  0.1136364 0.8863636 0.0588235 0.1481481
## [Resample] iter 238:  0.1818182 0.8181818 0.1764706 0.1851852
## [Resample] iter 239:  0.1136364 0.8863636 0.0588235 0.1481481
## [Resample] iter 240:  0.0681818 0.9318182 0.0588235 0.0740741
## [Resample] iter 241:  0.0454545 0.9545455 0.0000000 0.0740741
## [Resample] iter 242:  0.1136364 0.8863636 0.0588235 0.1481481
## [Resample] iter 243:  0.0465116 0.9534884 0.0000000 0.0740741
## [Resample] iter 244:  0.0697674 0.9302326 0.0000000 0.1153846
## [Resample] iter 245:  0.1363636 0.8636364 0.1176471 0.1481481
## [Resample] iter 246:  0.2045455 0.7954545 0.2352941 0.1851852
## [Resample] iter 247:  0.1190476 0.8809524 0.1875000 0.0769231
## [Resample] iter 248:  0.0909091 0.9090909 0.1176471 0.0740741
## [Resample] iter 249:  0.0465116 0.9534884 0.0000000 0.0769231
## [Resample] iter 250:  0.1136364 0.8863636 0.1176471 0.1111111
## [Resample] iter 251:  0.1395349 0.8604651 0.1764706 0.1153846
## [Resample] iter 252:  0.1162791 0.8837209 0.0625000 0.1481481
## [Resample] iter 253:  0.0697674 0.9302326 0.0588235 0.0769231
## [Resample] iter 254:  0.0909091 0.9090909 0.0588235 0.1111111
## [Resample] iter 255:  0.0681818 0.9318182 0.0588235 0.0740741
## [Resample] iter 256:  0.0697674 0.9302326 0.0625000 0.0740741
## [Resample] iter 257:  0.1136364 0.8863636 0.0588235 0.1481481
## [Resample] iter 258:  0.1395349 0.8604651 0.0588235 0.1923077
## [Resample] iter 259:  0.0909091 0.9090909 0.0588235 0.1111111
## [Resample] iter 260:  0.0909091 0.9090909 0.1764706 0.0370370
## [Resample] iter 261:  0.0909091 0.9090909 0.0000000 0.1481481
```

```
## [Resample] iter 262:  0.0697674 0.9302326 0.0000000 0.1153846
## [Resample] iter 263:  0.1136364 0.8863636 0.1176471 0.1111111
## [Resample] iter 264:  0.1136364 0.8863636 0.1176471 0.1111111
## [Resample] iter 265:  0.0465116 0.9534884 0.1250000 0.0000000
## [Resample] iter 266:  0.1904762 0.8095238 0.1250000 0.2307692
## [Resample] iter 267:  0.0681818 0.9318182 0.1176471 0.0370370
## [Resample] iter 268:  0.1395349 0.8604651 0.1176471 0.1538462
## [Resample] iter 269:  0.0227273 0.9772727 0.0000000 0.0370370
## [Resample] iter 270:  0.1363636 0.8636364 0.1176471 0.1481481
## [Resample] iter 271:  0.0909091 0.9090909 0.1176471 0.0740741
## [Resample] iter 272:  0.0681818 0.9318182 0.0588235 0.0740741
## [Resample] iter 273:  0.0909091 0.9090909 0.0000000 0.1481481
## [Resample] iter 274:  0.0930233 0.9069767 0.1875000 0.0370370
## [Resample] iter 275:  0.0454545 0.9545455 0.0588235 0.0370370
## [Resample] iter 276:  0.1395349 0.8604651 0.0588235 0.1923077
## [Resample] iter 277:  0.1162791 0.8837209 0.0588235 0.1538462
## [Resample] iter 278:  0.1162791 0.8837209 0.1764706 0.0769231
## [Resample] iter 279:  0.1162791 0.8837209 0.0625000 0.1481481
## [Resample] iter 280:  0.0909091 0.9090909 0.0588235 0.1111111
## [Resample] iter 281:  0.0697674 0.9302326 0.1250000 0.0370370
## [Resample] iter 282:  0.0681818 0.9318182 0.0588235 0.0740741
## [Resample] iter 283:  0.0697674 0.9302326 0.0000000 0.1111111
## [Resample] iter 284:  0.1363636 0.8636364 0.0588235 0.1851852
## [Resample] iter 285:  0.0227273 0.9772727 0.0000000 0.0370370
## [Resample] iter 286:  0.1395349 0.8604651 0.0588235 0.1923077
## [Resample] iter 287:  0.1162791 0.8837209 0.0588235 0.1538462
## [Resample] iter 288:  0.1363636 0.8636364 0.2352941 0.0740741
## [Resample] iter 289:  0.0930233 0.9069767 0.0588235 0.1153846
## [Resample] iter 290:  0.1363636 0.8636364 0.1764706 0.1111111
## [Resample] iter 291:  0.0227273 0.9772727 0.0588235 0.0000000
## [Resample] iter 292:  0.1627907 0.8372093 0.2500000 0.1111111
## [Resample] iter 293:  0.1363636 0.8636364 0.2352941 0.0740741
## [Resample] iter 294:  0.2558140 0.7441860 0.1176471 0.3461538
## [Resample] iter 295:  0.0930233 0.9069767 0.0000000 0.1538462
## [Resample] iter 296:  0.0697674 0.9302326 0.0625000 0.0740741
## [Resample] iter 297:  0.1395349 0.8604651 0.1764706 0.1153846
```

```
## [Resample] iter 298:  0.0454545 0.9545455 0.0000000 0.0740741
## [Resample] iter 299:  0.0454545 0.9545455 0.0000000 0.0740741
## [Resample] iter 300:  0.0454545 0.9545455 0.0000000 0.0740741
## [Resample] iter 301:  0.1395349 0.8604651 0.2352941 0.0769231
## [Resample] iter 302:  0.0697674 0.9302326 0.0625000 0.0740741
## [Resample] iter 303:  0.0681818 0.9318182 0.0588235 0.0740741
## [Resample] iter 304:  0.0681818 0.9318182 0.0588235 0.0740741
## [Resample] iter 305:  0.1428571 0.8571429 0.0625000 0.1923077
## [Resample] iter 306:  0.1162791 0.8837209 0.0588235 0.1538462
## [Resample] iter 307:  0.0681818 0.9318182 0.0000000 0.1111111
## [Resample] iter 308:  0.0454545 0.9545455 0.0588235 0.0370370
## [Resample] iter 309:  0.0454545 0.9545455 0.0000000 0.0740741
## [Resample] iter 310:  0.2272727 0.7727273 0.2352941 0.2222222
## [Resample] iter 311:  0.0697674 0.9302326 0.1176471 0.0384615
## [Resample] iter 312:  0.1363636 0.8636364 0.1176471 0.1481481
## [Resample] iter 313:  0.0465116 0.9534884 0.0000000 0.0740741
## [Resample] iter 314:  0.0909091 0.9090909 0.0588235 0.1111111
## [Resample] iter 315:  0.1363636 0.8636364 0.1176471 0.1481481
## [Resample] iter 316:  0.0232558 0.9767442 0.0000000 0.0384615
## [Resample] iter 317:  0.0476190 0.9523810 0.0625000 0.0384615
## [Resample] iter 318:  0.0909091 0.9090909 0.0588235 0.1111111
## [Resample] iter 319:  0.2500000 0.7500000 0.2352941 0.2592593
## [Resample] iter 320:  0.0909091 0.9090909 0.0588235 0.1111111
## [Resample] iter 321:  0.1860465 0.8139535 0.1250000 0.2222222
## [Resample] iter 322:  0.0681818 0.9318182 0.0588235 0.0740741
## [Resample] iter 323:  0.0227273 0.9772727 0.0588235 0.0000000
## [Resample] iter 324:  0.1395349 0.8604651 0.1176471 0.1538462
## [Resample] iter 325:  0.0232558 0.9767442 0.0000000 0.0370370
## [Resample] iter 326:  0.0909091 0.9090909 0.0588235 0.1111111
## [Resample] iter 327:  0.1363636 0.8636364 0.1764706 0.1111111
## [Resample] iter 328:  0.1136364 0.8863636 0.0588235 0.1481481
## [Resample] iter 329:  0.0697674 0.9302326 0.0588235 0.0769231
## [Resample] iter 330:  0.1395349 0.8604651 0.1176471 0.1538462
## [Resample] iter 331:  0.0454545 0.9545455 0.0588235 0.0370370
## [Resample] iter 332:  0.1162791 0.8837209 0.0588235 0.1538462
## [Resample] iter 333:  0.0952381 0.9047619 0.0625000 0.1153846
```

```
## [Resample] iter 334:  0.1136364 0.8863636 0.0000000 0.1851852
## [Resample] iter 335:  0.1818182 0.8181818 0.1176471 0.2222222
## [Resample] iter 336:  0.1136364 0.8863636 0.1176471 0.1111111
## [Resample] iter 337:  0.1162791 0.8837209 0.1764706 0.0769231
## [Resample] iter 338:  0.0454545 0.9545455 0.0000000 0.0740741
## [Resample] iter 339:  0.0454545 0.9545455 0.0588235 0.0370370
## [Resample] iter 340:  0.1162791 0.8837209 0.1875000 0.0740741
## [Resample] iter 341:  0.0952381 0.9047619 0.1250000 0.0769231
## [Resample] iter 342:  0.0697674 0.9302326 0.0588235 0.0769231
## [Resample] iter 343:  0.1590909 0.8409091 0.0588235 0.2222222
## [Resample] iter 344:  0.0930233 0.9069767 0.0625000 0.1111111
## [Resample] iter 345:  0.0909091 0.9090909 0.0588235 0.1111111
## [Resample] iter 346:  0.0681818 0.9318182 0.0588235 0.0740741
## [Resample] iter 347:  0.0697674 0.9302326 0.0588235 0.0769231
## [Resample] iter 348:  0.1590909 0.8409091 0.0588235 0.2222222
## [Resample] iter 349:  0.0454545 0.9545455 0.0588235 0.0370370
## [Resample] iter 350:  0.1136364 0.8863636 0.1764706 0.0740741
## [Resample] iter 351:  0.1136364 0.8863636 0.0000000 0.1851852
## [Resample] iter 352:  0.0476190 0.9523810 0.0000000 0.0769231
## [Resample] iter 353:  0.0930233 0.9069767 0.1176471 0.0769231
## [Resample] iter 354:  0.0909091 0.9090909 0.1176471 0.0740741
## [Resample] iter 355:  0.1627907 0.8372093 0.1764706 0.1538462
## [Resample] iter 356:  0.1136364 0.8863636 0.0588235 0.1481481
## [Resample] iter 357:  0.1136364 0.8863636 0.1764706 0.0740741
## [Resample] iter 358:  0.0930233 0.9069767 0.1250000 0.0740741
## [Resample] iter 359:  0.0681818 0.9318182 0.0000000 0.1111111
## [Resample] iter 360:  0.1136364 0.8863636 0.1176471 0.1111111
## [Resample] iter 361:  0.0681818 0.9318182 0.0588235 0.0740741
## [Resample] iter 362:  0.0697674 0.9302326 0.1176471 0.0384615
## [Resample] iter 363:  0.0909091 0.9090909 0.0588235 0.1111111
## [Resample] iter 364:  0.1136364 0.8863636 0.0588235 0.1481481
## [Resample] iter 365:  0.1395349 0.8604651 0.1176471 0.1538462
## [Resample] iter 366:  0.0681818 0.9318182 0.0000000 0.1111111
## [Resample] iter 367:  0.0930233 0.9069767 0.1250000 0.0740741
## [Resample] iter 368:  0.1428571 0.8571429 0.1250000 0.1538462
## [Resample] iter 369:  0.1363636 0.8636364 0.0588235 0.1851852
```

```
## [Resample] iter 370:  0.0681818 0.9318182 0.1176471 0.0370370
## [Resample] iter 371:  0.0227273 0.9772727 0.0000000 0.0370370
## [Resample] iter 372:  0.0909091 0.9090909 0.0588235 0.1111111
## [Resample] iter 373:  0.1666667 0.8333333 0.0625000 0.2307692
## [Resample] iter 374:  0.1162791 0.8837209 0.1250000 0.1111111
## [Resample] iter 375:  0.1162791 0.8837209 0.2941176 0.0000000
## [Resample] iter 376:  0.0681818 0.9318182 0.0588235 0.0740741
## [Resample] iter 377:  0.1818182 0.8181818 0.0000000 0.2962963
## [Resample] iter 378:  0.1395349 0.8604651 0.1176471 0.1538462
## [Resample] iter 379:  0.0454545 0.9545455 0.0588235 0.0370370
## [Resample] iter 380:  0.0227273 0.9772727 0.0000000 0.0370370
## [Resample] iter 381:  0.1136364 0.8863636 0.1176471 0.1111111
## [Resample] iter 382:  0.0227273 0.9772727 0.0588235 0.0000000
## [Resample] iter 383:  0.0930233 0.9069767 0.1176471 0.0769231
## [Resample] iter 384:  0.1860465 0.8139535 0.1250000 0.2222222
## [Resample] iter 385:  0.0909091 0.9090909 0.0588235 0.1111111
## [Resample] iter 386:  0.0714286 0.9285714 0.1250000 0.0384615
## [Resample] iter 387:  0.0909091 0.9090909 0.0000000 0.1481481
## [Resample] iter 388:  0.1395349 0.8604651 0.1176471 0.1538462
## [Resample] iter 389:  0.0909091 0.9090909 0.0000000 0.1481481
## [Resample] iter 390:  0.0681818 0.9318182 0.0588235 0.0740741
## [Resample] iter 391:  0.1136364 0.8863636 0.1764706 0.0740741
## [Resample] iter 392:  0.0465116 0.9534884 0.0000000 0.0769231
## [Resample] iter 393:  0.1590909 0.8409091 0.1176471 0.1851852
## [Resample] iter 394:  0.0697674 0.9302326 0.1875000 0.0000000
## [Resample] iter 395:  0.1818182 0.8181818 0.1764706 0.1851852
## [Resample] iter 396:  0.0697674 0.9302326 0.0000000 0.1111111
## [Resample] iter 397:  0.0232558 0.9767442 0.0588235 0.0000000
## [Resample] iter 398:  0.0930233 0.9069767 0.0000000 0.1538462
## [Resample] iter 399:  0.0909091 0.9090909 0.0588235 0.1111111
## [Resample] iter 400:  0.1363636 0.8636364 0.0588235 0.1851852
## [Resample] iter 401:  0.0681818 0.9318182 0.0588235 0.0740741
## [Resample] iter 402:  0.0930233 0.9069767 0.1250000 0.0740741
## [Resample] iter 403:  0.0930233 0.9069767 0.0588235 0.1153846
## [Resample] iter 404:  0.1162791 0.8837209 0.1176471 0.1153846
## [Resample] iter 405:  0.0909091 0.9090909 0.0000000 0.1481481
```

```
## [Resample] iter 406:  0.1162791 0.8837209 0.0625000 0.1481481
## [Resample] iter 407:  0.1590909 0.8409091 0.1176471 0.1851852
## [Resample] iter 408:  0.1136364 0.8863636 0.1764706 0.0740741
## [Resample] iter 409:  0.0454545 0.9545455 0.0588235 0.0370370
## [Resample] iter 410:  0.0697674 0.9302326 0.0000000 0.1153846
## [Resample] iter 411:  0.0465116 0.9534884 0.0625000 0.0370370
## [Resample] iter 412:  0.0681818 0.9318182 0.0588235 0.0740741
## [Resample] iter 413:  0.1363636 0.8636364 0.1176471 0.1481481
## [Resample] iter 414:  0.1395349 0.8604651 0.0625000 0.1851852
## [Resample] iter 415:  0.0454545 0.9545455 0.0588235 0.0370370
## [Resample] iter 416:  0.0681818 0.9318182 0.0588235 0.0740741
## [Resample] iter 417:  0.1162791 0.8837209 0.1176471 0.1153846
## [Resample] iter 418:  0.1627907 0.8372093 0.0588235 0.2307692
## [Resample] iter 419:  0.1363636 0.8636364 0.2352941 0.0740741
## [Resample] iter 420:  0.0697674 0.9302326 0.0000000 0.1153846
## [Resample] iter 421:  0.1590909 0.8409091 0.1176471 0.1851852
## [Resample] iter 422:  0.1363636 0.8636364 0.1764706 0.1111111
## [Resample] iter 423:  0.1136364 0.8863636 0.1176471 0.1111111
## [Resample] iter 424:  0.1162791 0.8837209 0.0588235 0.1538462
## [Resample] iter 425:  0.0681818 0.9318182 0.0588235 0.0740741
## [Resample] iter 426:  0.0227273 0.9772727 0.0000000 0.0370370
## [Resample] iter 427:  0.0930233 0.9069767 0.1250000 0.0740741
## [Resample] iter 428:  0.0697674 0.9302326 0.0588235 0.0769231
## [Resample] iter 429:  0.0952381 0.9047619 0.0625000 0.1153846
## [Resample] iter 430:  0.1136364 0.8863636 0.0588235 0.1481481
## [Resample] iter 431:  0.0465116 0.9534884 0.0588235 0.0384615
## [Resample] iter 432:  0.0930233 0.9069767 0.1176471 0.0769231
## [Resample] iter 433:  0.1627907 0.8372093 0.1250000 0.1851852
## [Resample] iter 434:  0.0909091 0.9090909 0.0588235 0.1111111
## [Resample] iter 435:  0.1590909 0.8409091 0.2352941 0.1111111
## [Resample] iter 436:  0.0909091 0.9090909 0.0588235 0.1111111
## [Resample] iter 437:  0.0465116 0.9534884 0.0000000 0.0769231
## [Resample] iter 438:  0.1363636 0.8636364 0.1176471 0.1481481
## [Resample] iter 439:  0.1162791 0.8837209 0.0000000 0.1851852
## [Resample] iter 440:  0.0454545 0.9545455 0.0588235 0.0370370
## [Resample] iter 441:  0.0227273 0.9772727 0.0000000 0.0370370
```

```
## [Resample] iter 442:  0.1395349 0.8604651 0.1764706 0.1153846
## [Resample] iter 443:  0.1162791 0.8837209 0.1764706 0.0769231
## [Resample] iter 444:  0.1590909 0.8409091 0.1176471 0.1851852
## [Resample] iter 445:  0.0227273 0.9772727 0.0000000 0.0370370
## [Resample] iter 446:  0.0909091 0.9090909 0.0588235 0.1111111
## [Resample] iter 447:  0.1627907 0.8372093 0.1176471 0.1923077
## [Resample] iter 448:  0.0930233 0.9069767 0.1250000 0.0740741
## [Resample] iter 449:  0.1162791 0.8837209 0.0625000 0.1481481
## [Resample] iter 450:  0.0681818 0.9318182 0.0000000 0.1111111
## [Resample] iter 451:  0.1590909 0.8409091 0.0000000 0.2592593
## [Resample] iter 452:  0.0697674 0.9302326 0.1176471 0.0384615
## [Resample] iter 453:  0.0952381 0.9047619 0.0625000 0.1153846
## [Resample] iter 454:  0.0930233 0.9069767 0.0588235 0.1153846
## [Resample] iter 455:  0.0909091 0.9090909 0.1176471 0.0740741
## [Resample] iter 456:  0.1136364 0.8863636 0.1176471 0.1111111
## [Resample] iter 457:  0.2045455 0.7954545 0.1764706 0.2222222
## [Resample] iter 458:  0.0465116 0.9534884 0.0625000 0.0370370
## [Resample] iter 459:  0.0454545 0.9545455 0.1176471 0.0000000
## [Resample] iter 460:  0.0909091 0.9090909 0.0000000 0.1481481
## [Resample] iter 461:  0.0909091 0.9090909 0.0000000 0.1481481
## [Resample] iter 462:  0.1136364 0.8863636 0.0000000 0.1851852
## [Resample] iter 463:  0.0681818 0.9318182 0.0000000 0.1111111
## [Resample] iter 464:  0.1136364 0.8863636 0.1764706 0.0740741
## [Resample] iter 465:  0.0227273 0.9772727 0.0588235 0.0000000
## [Resample] iter 466:  0.0681818 0.9318182 0.0588235 0.0740741
## [Resample] iter 467:  0.0930233 0.9069767 0.1176471 0.0769231
## [Resample] iter 468:  0.1666667 0.8333333 0.2500000 0.1153846
## [Resample] iter 469:  0.0930233 0.9069767 0.0000000 0.1538462
## [Resample] iter 470:  0.0930233 0.9069767 0.1250000 0.0740741
## [Resample] iter 471:  0.0454545 0.9545455 0.0588235 0.0370370
## [Resample] iter 472:  0.0232558 0.9767442 0.0588235 0.0000000
## [Resample] iter 473:  0.2045455 0.7954545 0.1764706 0.2222222
## [Resample] iter 474:  0.1428571 0.8571429 0.0625000 0.1923077
## [Resample] iter 475:  0.0681818 0.9318182 0.0000000 0.1111111
## [Resample] iter 476:  0.1860465 0.8139535 0.1250000 0.2222222
## [Resample] iter 477:  0.0227273 0.9772727 0.0000000 0.0370370
```

```
## [Resample] iter 478:  0.1162791 0.8837209 0.1176471 0.1153846
## [Resample] iter 479:  0.1136364 0.8863636 0.1176471 0.1111111
## [Resample] iter 480:  0.0681818 0.9318182 0.1176471 0.0370370
## [Resample] iter 481:  0.0227273 0.9772727 0.0000000 0.0370370
## [Resample] iter 482:  0.0909091 0.9090909 0.1176471 0.0740741
## [Resample] iter 483:  0.1818182 0.8181818 0.1764706 0.1851852
## [Resample] iter 484:  0.0465116 0.9534884 0.0588235 0.0384615
## [Resample] iter 485:  0.1162791 0.8837209 0.0000000 0.1923077
## [Resample] iter 486:  0.0909091 0.9090909 0.0588235 0.1111111
## [Resample] iter 487:  0.0476190 0.9523810 0.0625000 0.0384615
## [Resample] iter 488:  0.1395349 0.8604651 0.1250000 0.1481481
## [Resample] iter 489:  0.1136364 0.8863636 0.0000000 0.1851852
## [Resample] iter 490:  0.1136364 0.8863636 0.1764706 0.0740741
## [Resample] iter 491:  0.0454545 0.9545455 0.0000000 0.0740741
## [Resample] iter 492:  0.1395349 0.8604651 0.0625000 0.1851852
## [Resample] iter 493:  0.0697674 0.9302326 0.0588235 0.0769231
## [Resample] iter 494:  0.0465116 0.9534884 0.0588235 0.0384615
## [Resample] iter 495:  0.1136364 0.8863636 0.1176471 0.1111111
## [Resample] iter 496:  0.0681818 0.9318182 0.1176471 0.0370370
## [Resample] iter 497:  0.1860465 0.8139535 0.1875000 0.1851852
## [Resample] iter 498:  0.0681818 0.9318182 0.1176471 0.0370370
## [Resample] iter 499:  0.1818182 0.8181818 0.0588235 0.2592593
## [Resample] iter 500:  0.0697674 0.9302326 0.0588235 0.0769231
##
## Aggregated Result: mmce.test.mean=0.0984480,acc.test.mean=0.9015520,fpr.test.mean=0.0825368,fnr.test
##
```

Prediction rate in accuracy (acc.)

```
bayesCV$aggr
```

```
## mmce.test.mean  acc.test.mean  fpr.test.mean  fnr.test.mean
##     0.09844800     0.90155200     0.08253676     0.10845014
```

If we want to predict the political party of a new politician (not in original data).

```
politician <- tibble(V1="n",V2="n",V3="y",V4="n",V5="n",V6V="y",V7V="y",V8="y",V9="y",V10="y",V11="n",V
politicianpred <- predict(bayesModel, newdata = politician)
```

```
## Warning in predict.WrappedModel(bayesModel, newdata = politician): Provided data
## for prediction is not a pure data.frame but from class tbl_df, hence it will be
## converted.
```

```
getPredictionResponse(politicianpred)
```

```
## [1] democrat
## Levels: democrat republican
```

Now if we wrap our model inside getLearnerModel() function. What are the prior and likelihood probabilities?

```
getLearnerModel(bayesModel)
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##   democrat republican
##  0.6137931  0.3862069
##
## Conditional probabilities:
##             V1
## Y                    n         y
##   democrat   0.3953488 0.6046512
##   republican 0.8121212 0.1878788
##
##             V2
## Y                    n         y
##   democrat   0.4979079 0.5020921
##   republican 0.4932432 0.5067568
##
##             V3
## Y                    n         y
##   democrat   0.1115385 0.8884615
##   republican 0.8658537 0.1341463
##
##             V4
## Y                     n          y
##   democrat   0.94594595 0.05405405
##   republican 0.01212121 0.98787879
##
##             V5
## Y                     n          y
##   democrat   0.78431373 0.21568627
##   republican 0.04848485 0.95151515
##
##             V6
## Y                    n         y
##   democrat   0.5232558 0.4767442
##   republican 0.1024096 0.8975904
##
##             V7
## Y                    n         y
##   democrat   0.2277992 0.7722008
##   republican 0.7592593 0.2407407
```

```
##
##             V8
## Y                    n         y
##   democrat   0.1711027 0.8288973
##   republican 0.8471338 0.1528662
##
##             V9
## Y                    n         y
##   democrat   0.2419355 0.7580645
##   republican 0.8848485 0.1151515
##
##             V10
## Y                    n         y
##   democrat   0.5285171 0.4714829
##   republican 0.4424242 0.5575758
##
##             V11
## Y                    n         y
##   democrat   0.4941176 0.5058824
##   republican 0.8679245 0.1320755
##
##             V12
## Y                    n         y
##   democrat   0.8554217 0.1445783
##   republican 0.1290323 0.8709677
##
##             V13
## Y                    n         y
##   democrat   0.7103175 0.2896825
##   republican 0.1392405 0.8607595
##
##             V14
## Y                     n          y
##   democrat   0.64980545 0.35019455
##   republican 0.01863354 0.98136646
##
##             V15
## Y                     n          y
##   democrat   0.36254980 0.63745020
##   republican 0.91025641 0.08974359
##
##             V16
## Y                     n          y
##   democrat   0.06486486 0.93513514
##   republican 0.34246575 0.65753425
```

Strengths and Weaknesses

S: continuous and categorical variables, inexpensive computanioally, no hyperparamaters to tune, handle cases of missing data.\ W: Assumes continuous predictors variables distribute normal, that they are iid.

SVM Algorithm

To build a classifier. Veremos un plano que separa los 1s de los 0s.\ Puede agregar una dimension mas para darnos el mejor hiperplano.Buenos para non-linear separable classes.\ SVM encuentra el hiperplano lineal optimo. (2 datos, el hiperplano es una linea, etc.). El algoritmo buscara el plano optimo que maximiza el margen alrededor del plano. El margen es la distancia alrededor del plano que toca el menor numero de casos. Los casos que tocan el margen se llaman SV. Nos ayudan a definir el boundary entre clases. (Ej. x = horas jugadas, y = money made, predict si boss bien o mal).\ Usaremos en muchas ocaciones "soft" margins que permiten algunos errores. Es un sacrificio de bias-variance contra los hard margins.

Non-linear data anda SVM

Al crear la dimension extra para nuestros datos (kernel), SVM puede separar no solo de formal lineal. El modelo usa un kernel-function (linear, polynomial, gaussian radial, sigmoid). Nosotros decidimos el tipo de funcion a usar.
Hiperparametros del SVM Kernel. degree (controla que tan bendy el limite de decision), cost (C, hard o soft margins), gamma (que tanta influencia tiene cada caso individual). Gamma alto puede dar overfitting y viceversa. ## More than 2 classes One-versus-all: tantos SVM models como clases; One-versus-one. Cuando no sirven para algunas clases, se usa Platt scaling.\

Model SVM

```
## Warning: `as.tibble()` was deprecated in tibble 2.0.0.
## Please use `as_tibble()` instead.
## The signature and semantics have changed, see `?as_tibble`.
```

```
## # A tibble: 6 x 58
##    make address   all num3d   our  over remove internet order  mail receive
##   <dbl>   <dbl> <dbl> <dbl> <dbl> <dbl>  <dbl>    <dbl> <dbl> <dbl>   <dbl>
## 1  0       0.64  0.64     0  0.32  0      0        0     0     0       0
## 2  0.21    0.28  0.5      0  0.14  0.28   0.21     0.07  0     0.94    0.21
## 3  0.06    0     0.71     0  1.23  0.19   0.19     0.12  0.64  0.25    0.38
## 4  0       0     0        0  0.63  0      0.31     0.63  0.31  0.63    0.31
## 5  0       0     0        0  0.63  0      0.31     0.63  0.31  0.63    0.31
## 6  0       0     0        0  1.85  0      0        1.85  0     0       0
## # ... with 47 more variables: will <dbl>, people <dbl>, report <dbl>,
## #   addresses <dbl>, free <dbl>, business <dbl>, email <dbl>, you <dbl>,
## #   credit <dbl>, your <dbl>, font <dbl>, num000 <dbl>, money <dbl>, hp <dbl>,
## #   hpl <dbl>, george <dbl>, num650 <dbl>, lab <dbl>, labs <dbl>, telnet <dbl>,
## #   num857 <dbl>, data <dbl>, num415 <dbl>, num85 <dbl>, technology <dbl>,
## #   num1999 <dbl>, parts <dbl>, pm <dbl>, direct <dbl>, cs <dbl>,
## #   meeting <dbl>, original <dbl>, project <dbl>, re <dbl>, edu <dbl>,
## #   table <dbl>, conference <dbl>, charSemicolon <dbl>, charRoundbracket <dbl>,
## #   charSquarebracket <dbl>, charExclamation <dbl>, charDollar <dbl>,
## #   charHash <dbl>, capitalAve <dbl>, capitalLong <dbl>, capitalTotal <dbl>,
## #   type <fct>
```

6.5.2 Tuning in parameters

Con esto sabremos que hiperparametros estan disponibles para tuning. Usamos getParamSet(). Nos dira cual es el valor definido, sino lo cambiamos; los constraints; si es necesario para el learner; si puede ser tuned. Do not forget to scale if necessary.

```
## Warning in makeTask(type = type, data = data, weights = weights, blocking =
## blocking, : Provided data is not a pure data.frame but from class tbl_df, hence
```

```
## it will be converted.
```

```
##                         Type  len             Def
## type                discrete    - C-classifica...
## cost                 numeric    -               1
## nu                   numeric    -             0.5
## class.weights numericvector <NA>               -
## kernel              discrete    -          radial
## degree               integer    -               3
## coef0                numeric    -               0
## gamma                numeric    -               -
## cachesize            numeric    -              40
## tolerance            numeric    -           0.001
## shrinking            logical    -            TRUE
## cross                integer    -               0
## fitted               logical    -            TRUE
## scale         logicalvector <NA>            TRUE
##                                          Constr Req Tunable Trafo
## type          C-classification,nu-classification   -    TRUE     -
## cost                                    0 to Inf   Y    TRUE     -
## nu                                   -Inf to Inf   Y    TRUE     -
## class.weights                           0 to Inf   -    TRUE     -
## kernel          linear,polynomial,radial,sigmoid   -    TRUE     -
## degree                                  1 to Inf   Y    TRUE     -
## coef0                                -Inf to Inf   Y    TRUE     -
## gamma                                   0 to Inf   Y    TRUE     -
## cachesize                            -Inf to Inf   -    TRUE     -
## tolerance                               0 to Inf   -    TRUE     -
## shrinking                                      -   -    TRUE     -
## cross                                   0 to Inf   -   FALSE     -
## fitted                                         -   -   FALSE     -
## scale                                          -   -    TRUE     -
```

```
##       Type len    Def                           Constr Req Tunable Trafo
## 1 discrete   - radial linear,polynomial,radial,sigmoid   -    TRUE     -
## Used: train.
```

```
##      Type len Def   Constr Req Tunable Trafo
## 1 integer   -   3 1 to Inf   Y    TRUE     -
## Used: train.
```

Consideramos los mas importantes: Kernel, Cost, Degree, Gamma. Enlistamos los kernels a probar. Definimos el espacio hiperparametrico.\ Kernel, valores discretos.\ Degree, valores enteros, definimos upper and lower values\ Gamma, numeric values, definimos upper and lower values.

It will find the best combination of the hyperparameter space. It is doing a grid-search. Para evitar no alargar el tiempo y carga computacional, usamos random search. Seleccion aleatoria de una combinacion de valores de hiperparametros; CV para entrenar y evaluar el modelo con esos valores; registrar el rendimiento del modelo; repetir los 3 pasos anteriores tanto como sea posible tiempo y dinero; seleccionar el mejor.\

```
randSearch <- makeTuneControlRandom(maxit = 20)
cvForTuning <- makeResampleDesc("Holdout", split = 2/3)
```

Busquemos correr el proceso en paralelo.

```
library(parallelMap)
parallelStartSocket(cpus = detectCores())
```

```
## Starting parallelization in mode=socket with cpus=8.
```

```
tunedSvmPars <- tuneParams("classif.svm", task = spamTask,
                     resampling = cvForTuning,
                     par.set = svmParamSpace,
                     control = randSearch)
```

```
## [Tune] Started tuning learner classif.svm for parameter set:
```

```
##            Type len Def                    Constr Req Tunable Trafo
## kernel discrete   -   - polynomial,radial,sigmoid   -    TRUE     -
## degree  integer   -   -                    1 to 3   -    TRUE     -
## cost    numeric   -   -                 0.1 to 10   -    TRUE     -
## gamma   numeric   -   -                 0.1 to 10   -    TRUE     -
```

```
## With control class: TuneControlRandom
```

```
## Imputation value: 1
```

```
## Exporting objects to slaves for mode socket: .mlr.slave.options
```

```
## Mapping in parallel: mode = socket; level = mlr.tuneParams; cpus = 8; elements = 20.
```

```
## [Tune] Result: kernel=polynomial; degree=1; cost=4.01; gamma=0.128 : mmce.test.mean=0.0606258
```

```
parallelStop()
```

```
## Stopped parallelization. All cleaned up.
```

Now we print the results.

```
tunedSvmPars
```

```
## Tune result:
## Op. pars: kernel=polynomial; degree=1; cost=4.01; gamma=0.128
## mmce.test.mean=0.0606258
```

```
tunedSvmPars$x
```

```
## $kernel
## [1] "polynomial"
##
## $degree
## [1] 1
##
## $cost
## [1] 4.005665
##
## $gamma
## [1] 0.1279402
```

6.5.3 Train model with hiperparameters

Usamos setHyperPars() para combinar learner con predefinidos valores de hiperparametros. pars.vals es el objeto que tiene nuestros valores hiperparametricos tuned.

```
tunedSvm <- setHyperPars(makeLearner("classif.svm"),
                         par.vals= tunedSvmPars$x)
tunedSvmModel <- train(tunedSvm, spamTask)
```

6.6 CV of SVM Model

No olvidar el uso de una wrapper function. Wrapps nuestro learner e hiperparametros tuning process. Primero definimos nuestra estrategia outer CV. Luego elegimos el resampling dentro de makeTuneWrapper(), que es el inner-loop CV. Siguiente, para correr nuestrto nested CV, usamos resample (primero wrapped learner, task, CV strategy).

```
outer <- makeResampleDesc("CV", iters = 3)

svmWrapper <- makeTuneWrapper("classif.svm", resampling = cvForTuning,par.set = svmParamSpace, control =

parallelStartSocket(cpus = detectCores())
```

```
## Starting parallelization in mode=socket with cpus=8.
```

```
cvWithTuning <- resample(svmWrapper, spamTask, resampling = outer)
```

```
## Exporting objects to slaves for mode socket: .mlr.slave.options
## Resampling: cross-validation
## Measures:             mmce
## Mapping in parallel: mode = socket; level = mlr.resample; cpus = 8; elements = 3.
##
## Aggregated Result: mmce.test.mean=0.0708556
##
```

```
cvWithTuning
```

```
## Resample Result
## Task: spamTib
## Learner: classif.svm.tuned
## Aggr perf: mmce.test.mean=0.0708556
## Runtime: 53.507
```

```
parallelStop()
```

```
## Stopped parallelization. All cleaned up.
```