rectangular_class_domain

Ye

7/28/2017

Load library

```
suppressWarnings(library(caret))
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

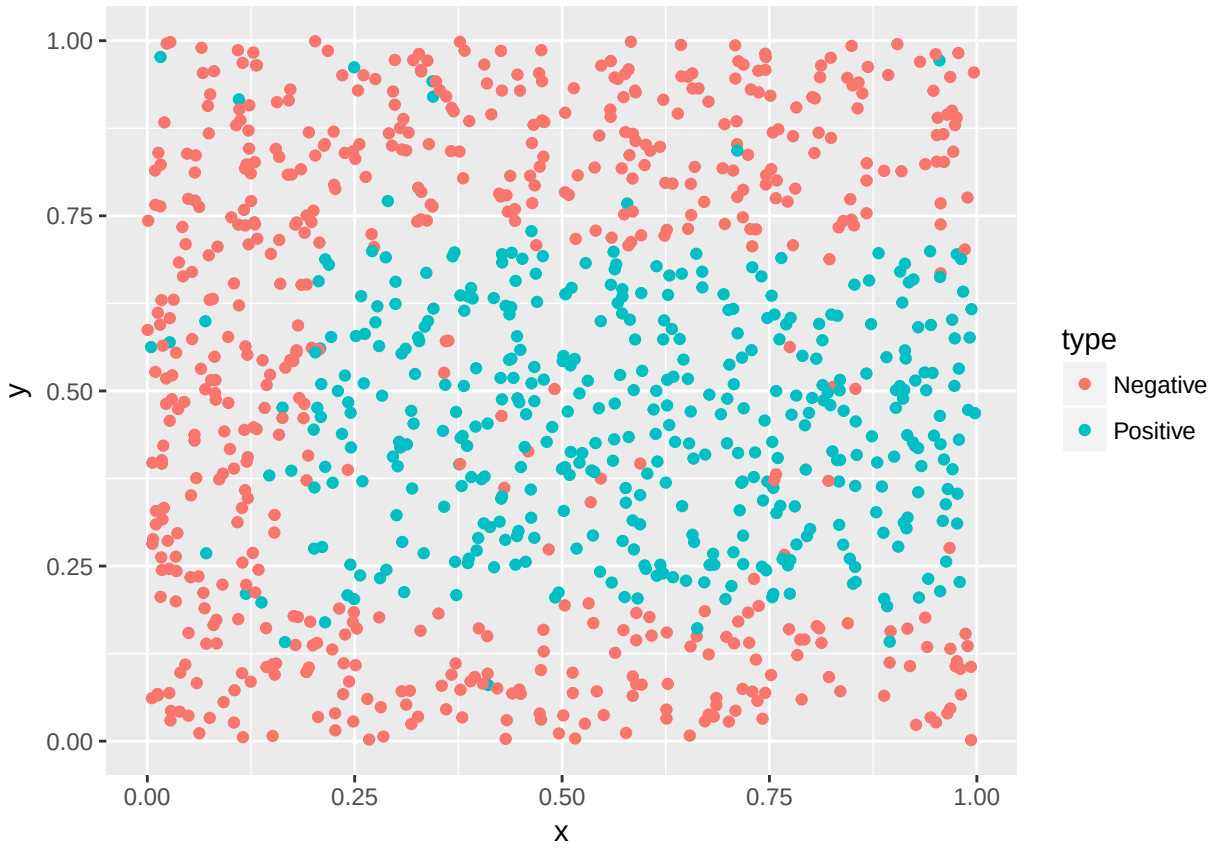
```
## Note: the specification for S3 class "family" in package 'MatrixModels' seems equivalent to one from
```

```
suppressWarnings(library(rpart))
```

Create rectangular class domain with certain randomness

```
N = 1000
xPos = 0.2
yMinPos = 0.2
yMaxPos = 0.7
newData = data.frame(x=runif(N),y=runif(N))
newData$type = with(newData,ifelse(x>xPos & y>yMinPos & y<yMaxPos,
                                   'Positive', 'Negative'))

n = N/10
newData$type[1:n] = c('Positive', 'Negative')[1+rbinom(n, 1, 0.5)]
newData$type = factor(newData$type)
newData = newData[sample(nrow(newData)),]
qplot(x=x,y=y,data=newData, color=type)
```



Logistic regression and perform cross validation (caret) to check the predictive quality

```
modelFormula = formula('type ~ x + y')
logrFit <- glm(modelFormula, family=binomial("logit"),data=newData)
print(summary(logrFit))
```

```
##
## Call:
## glm(formula = modelFormula, family = binomial("logit"), data = newData)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7688  -0.9701  -0.6655   1.1329   2.0628
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.9256     0.1798  -5.148 2.63e-07 ***
## x             2.2727     0.2445   9.294  < 2e-16 ***
## y            -1.1378     0.2433  -4.677 2.92e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 1355.9  on 999  degrees of freedom
## Residual deviance: 1241.1  on 997  degrees of freedom
## AIC: 1247.1
##
## Number of Fisher Scoring iterations: 4
```

```
ctrl <- trainControl(method = "cv", number = 10)
logrTrain <- train(modelFormula, data=newData,
                   method = 'glm', trControl = ctrl)
summary(logrTrain)
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7688  -0.9701  -0.6655   1.1329   2.0628
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.9256     0.1798  -5.148 2.63e-07 ***
## x             2.2727     0.2445   9.294  < 2e-16 ***
## y            -1.1378     0.2433  -4.677 2.92e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1355.9  on 999  degrees of freedom
## Residual deviance: 1241.1  on 997  degrees of freedom
## AIC: 1247.1
##
## Number of Fisher Scoring iterations: 4
```

Use classification tree to fit the data

```
treeFit <- rpart(modelFormula, data=newData)
printcp(treeFit)
```

```
##
## Classification tree:
## rpart(formula = modelFormula, data = newData)
##
## Variables actually used in tree construction:
## [1] x y
##
## Root node error: 413/1000 = 0.413
##
## n= 1000
##
##        CP nsplit rel error  xerror     xstd
```

```
## 1 0.32203      0   1.00000 1.00000 0.037700
## 2 0.23487      2   0.35593 0.36320 0.027340
## 3 0.01000      3   0.12107 0.12833 0.017154
```

```
treeTrain <- train(modelFormula, method="rpart",data=newData,
                   trControl = ctrl)
```

Compare the two methods

```
print(list(tree = treeTrain$results, logit = logrTrain$results))
```

```
## $tree
##          cp  Accuracy     Kappa AccuracySD    KappaSD
## 1 0.0000000 0.9419590 0.8801135 0.02706817 0.05596692
## 2 0.2348668 0.8859867 0.7727136 0.05704972 0.10920718
## 3 0.3220339 0.7599867 0.4713518 0.12047061 0.32633394
##
## $logit
##   parameter  Accuracy     Kappa AccuracySD   KappaSD
## 1      none 0.6349534 0.2234063 0.04865235 0.1020132
```