Support_Vector_Machine_loan

Nabil Momin

2024-06-10

```
library(corrgram)
library(corrplot)

## corrplot 0.92 loaded

library(caTools)
library(Amelia)

## Loading required package: Rcpp

## ##
## ## Amelia II: Multiple Imputation
## ## (Version 1.8.2, built: 2024-04-10)
## ## Copyright (C) 2005-2024 James Honaker, Gary King and Matthew Blackwell
## ## Refer to http://gking.harvard.edu/amelia/ for more information
## ##

library(ggplot2)
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

library(rpart)
library(rpart.plot)
library(randomForest)

## randomForest 4.7-1.1

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine

## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
library(ISLR)
library(e1071)
```

```
#### importing the csv file

loan <- read.csv('loan_data.csv')
```

```
View(loan)
```

```
str(loan)
```

```
## 'data.frame':    9578 obs. of  14 variables:
##  $ credit.policy    : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ purpose          : chr  "debt_consolidation" "credit_card"
"debt_consolidation" "debt_consolidation" "debt_consolidation" ...
##  $ int.rate         : num  0.119 0.107 0.136 0.101 0.143 ...
##  $ installment      : num  829 228 367 162 103 ...
##  $ log.annual.inc   : num  11.4 11.1 10.4 11.4 11.3 ...
##  $ dti              : num  19.5 14.3 11.6 8.1 15 ...
##  $ fico             : int  737 707 682 712 667 727 667 722 682 707 ...
##  $ days.with.cr.line: num  5640 2760 4710 2700 4066 ...
##  $ revol.bal        : int  28854 33623 3511 33667 4740 50807 3839 24220
69909 5630 ...
##  $ revol.util       : num  52.1 76.7 25.6 73.2 39.5 51 76.8 68.6 51.1 23
...
##  $ inq.last.6mths   : int  0 0 1 1 0 0 0 0 1 1 ...
##  $ delinq.2yrs      : int  0 0 0 0 1 0 0 0 0 0 ...
##  $ pub.rec          : int  0 0 0 0 0 0 1 0 0 0 ...
##  $ not.fully.paid   : int  0 0 0 0 0 0 1 1 0 0 ...
```

```
##### checking which column we can factor
```

```
str(loan)
```

```
## 'data.frame':    9578 obs. of  14 variables:
##  $ credit.policy    : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ purpose          : chr  "debt_consolidation" "credit_card"
"debt_consolidation" "debt_consolidation" "debt_consolidation" ...
##  $ int.rate         : num  0.119 0.107 0.136 0.101 0.143 ...
##  $ installment      : num  829 228 367 162 103 ...
##  $ log.annual.inc   : num  11.4 11.1 10.4 11.4 11.3 ...
##  $ dti              : num  19.5 14.3 11.6 8.1 15 ...
```

```
##  $ fico             : int  737 707 682 712 667 727 667 722 682 707 ...
##  $ days.with.cr.line: num  5640 2760 4710 2700 4066 ...
##  $ revol.bal        : int  28854 33623 3511 33667 4740 50807 3839 24220
69909 5630 ...
##  $ revol.util       : num  52.1 76.7 25.6 73.2 39.5 51 76.8 68.6 51.1 23
...
##  $ inq.last.6mths   : int  0 0 1 1 0 0 0 0 1 1 ...
##  $ delinq.2yrs      : int  0 0 0 0 1 0 0 0 0 0 ...
##  $ pub.rec          : int  0 0 0 0 0 0 1 0 0 0 ...
##  $ not.fully.paid   : int  0 0 0 0 0 0 1 1 0 0 ...
```

```
summary(loan)
```

```
##  credit.policy     purpose            int.rate       installment
##  Min.   :0.000   Length:9578        Min.   :0.0600   Min.   : 15.67
##  1st Qu.:1.000   Class :character   1st Qu.:0.1039   1st Qu.:163.77
##  Median :1.000   Mode  :character   Median :0.1221   Median :268.95
##  Mean   :0.805                      Mean   :0.1226   Mean   :319.09
##  3rd Qu.:1.000                      3rd Qu.:0.1407   3rd Qu.:432.76
##  Max.   :1.000                      Max.   :0.2164   Max.   :940.14
##  log.annual.inc        dti              fico       days.with.cr.line
##  Min.   : 7.548   Min.   : 0.000   Min.   :612.0   Min.   :  179
##  1st Qu.:10.558   1st Qu.: 7.213   1st Qu.:682.0   1st Qu.: 2820
##  Median :10.929   Median :12.665   Median :707.0   Median : 4140
##  Mean   :10.932   Mean   :12.607   Mean   :710.8   Mean   : 4561
##  3rd Qu.:11.291   3rd Qu.:17.950   3rd Qu.:737.0   3rd Qu.: 5730
##  Max.   :14.528   Max.   :29.960   Max.   :827.0   Max.   :17640
##    revol.bal         revol.util    inq.last.6mths    delinq.2yrs
##  Min.   :      0   Min.   :  0.0   Min.   : 0.000   Min.   : 0.0000
##  1st Qu.:   3187   1st Qu.: 22.6   1st Qu.: 0.000   1st Qu.: 0.0000
##  Median :   8596   Median : 46.3   Median : 1.000   Median : 0.0000
##  Mean   :  16914   Mean   : 46.8   Mean   : 1.577   Mean   : 0.1637
##  3rd Qu.:  18250   3rd Qu.: 70.9   3rd Qu.: 2.000   3rd Qu.: 0.0000
##  Max.   :1207359   Max.   :119.0   Max.   :33.000   Max.   :13.0000
##     pub.rec        not.fully.paid
##  Min.   :0.00000   Min.   :0.0000
##  1st Qu.:0.00000   1st Qu.:0.0000
##  Median :0.00000   Median :0.0000
##  Mean   :0.06212   Mean   :0.1601
##  3rd Qu.:0.00000   3rd Qu.:0.0000
##  Max.   :5.00000   Max.   :1.0000
```

```
table(loan$credit.policy)
```

```
##
##    0    1
## 1868 7710
```

```
####
```

```
loan$credit.policy <- factor(loan$credit.policy)
```

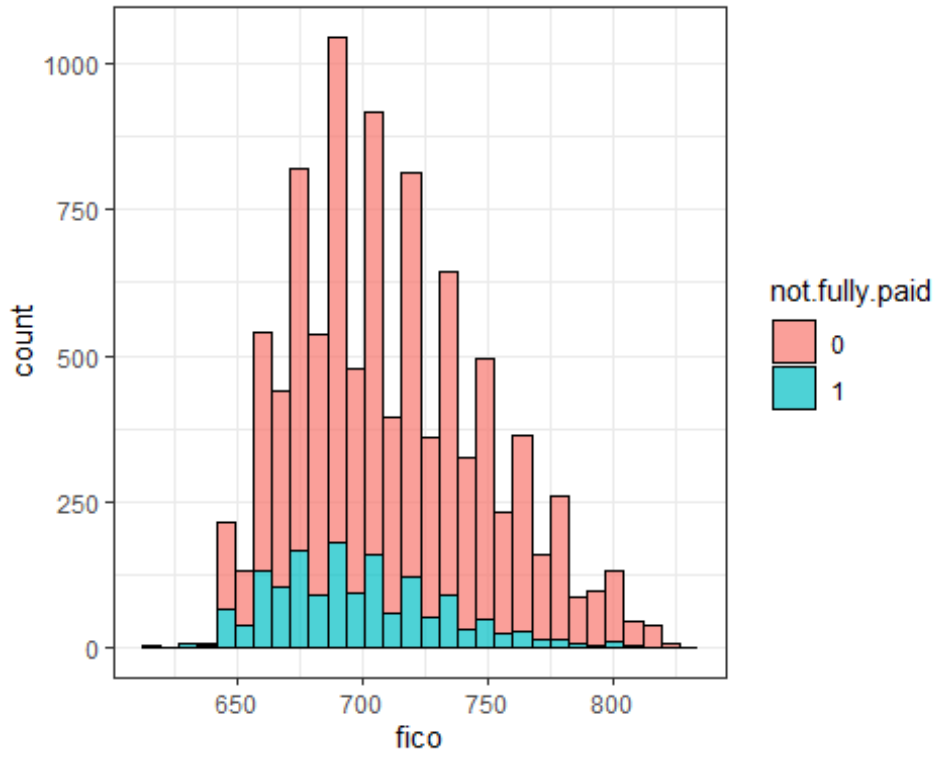
```r
loan$inq.last.6mths <- factor(loan$inq.last.6mths)
loan$delinq.2yrs <- factor(loan$delinq.2yrs)
loan$pub.rec <- factor(loan$pub.rec)
loan$not.fully.paid <- factor(loan$not.fully.paid)

str(loan)
```

```
## 'data.frame':    9578 obs. of  14 variables:
##  $ credit.policy    : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
##  $ purpose          : chr  "debt_consolidation" "credit_card"
"debt_consolidation" "debt_consolidation" "debt_consolidation" ...
##  $ int.rate         : num  0.119 0.107 0.136 0.101 0.143 ...
##  $ installment      : num  829 228 367 162 103 ...
##  $ log.annual.inc   : num  11.4 11.1 10.4 11.4 11.3 ...
##  $ dti              : num  19.5 14.3 11.6 8.1 15 ...
##  $ fico             : int  737 707 682 712 667 727 667 722 682 707 ...
##  $ days.with.cr.line: num  5640 2760 4710 2700 4066 ...
##  $ revol.bal        : int  28854 33623 3511 33667 4740 50807 3839 24220
69909 5630 ...
##  $ revol.util       : num  52.1 76.7 25.6 73.2 39.5 51 76.8 68.6 51.1 23
...
##  $ inq.last.6mths   : Factor w/ 28 levels "0","1","2","3",..: 1 1 2 2 1 1
1 1 2 2 ...
##  $ delinq.2yrs      : Factor w/ 11 levels "0","1","2","3",..: 1 1 1 1 2 1
1 1 1 1 ...
##  $ pub.rec          : Factor w/ 6 levels "0","1","2","3",..: 1 1 1 1 1 1 2
1 1 1 ...
##  $ not.fully.paid   : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 2 2 1 1 ...
```

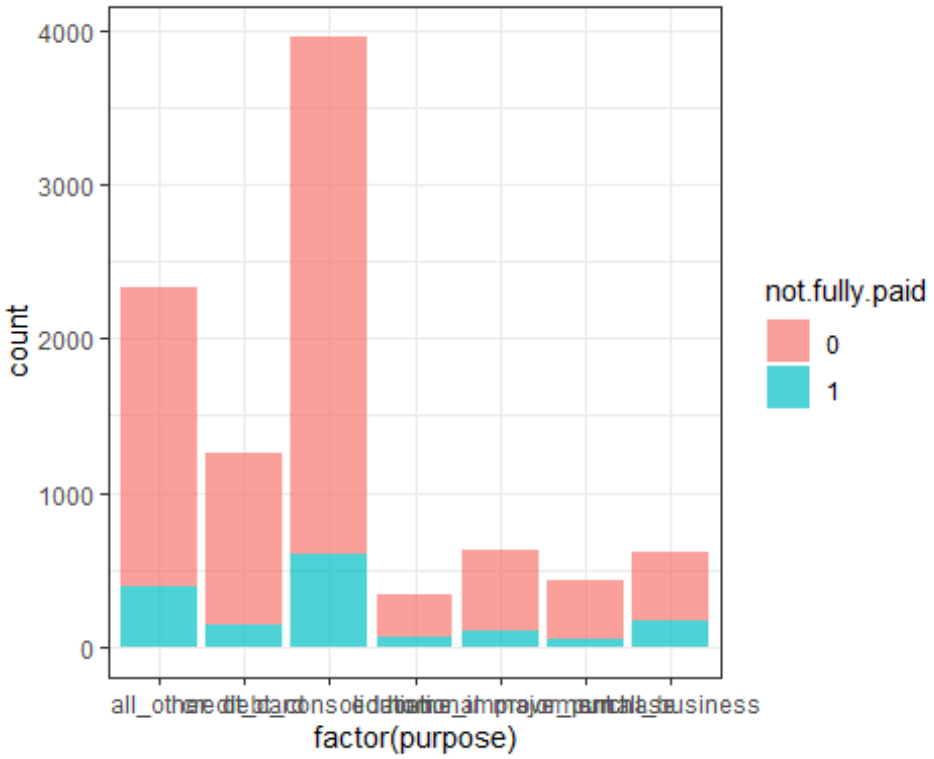
EDA time

```r
ggplot(loan,aes(fico)) +
geom_histogram(aes(fill=not.fully.paid),color='black',alpha=0.7) + theme_bw()
```

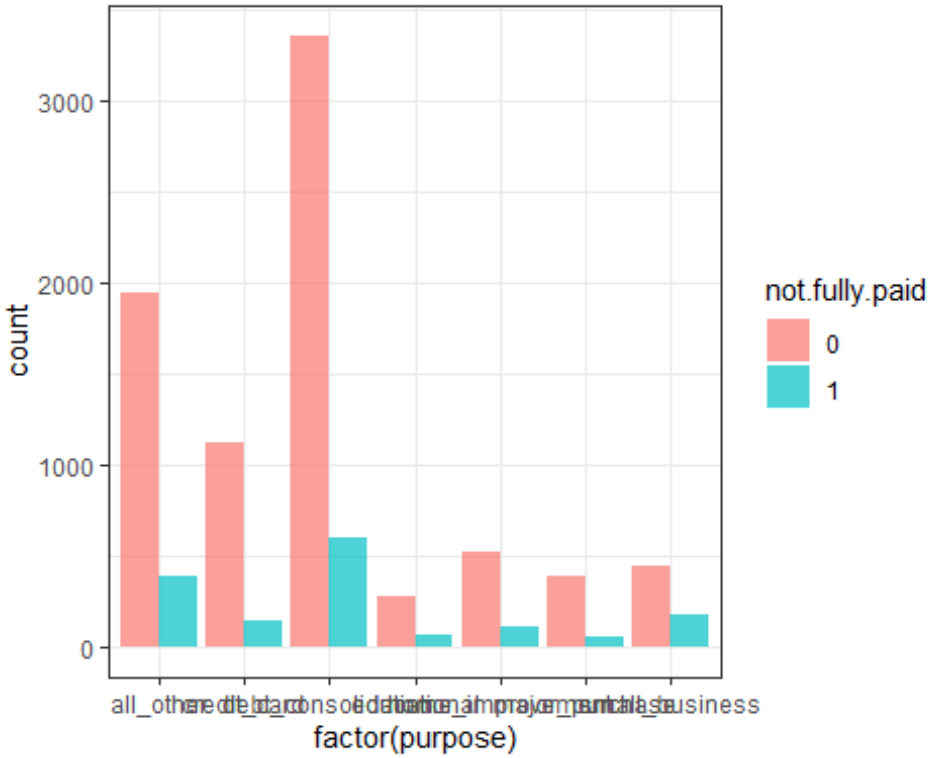
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



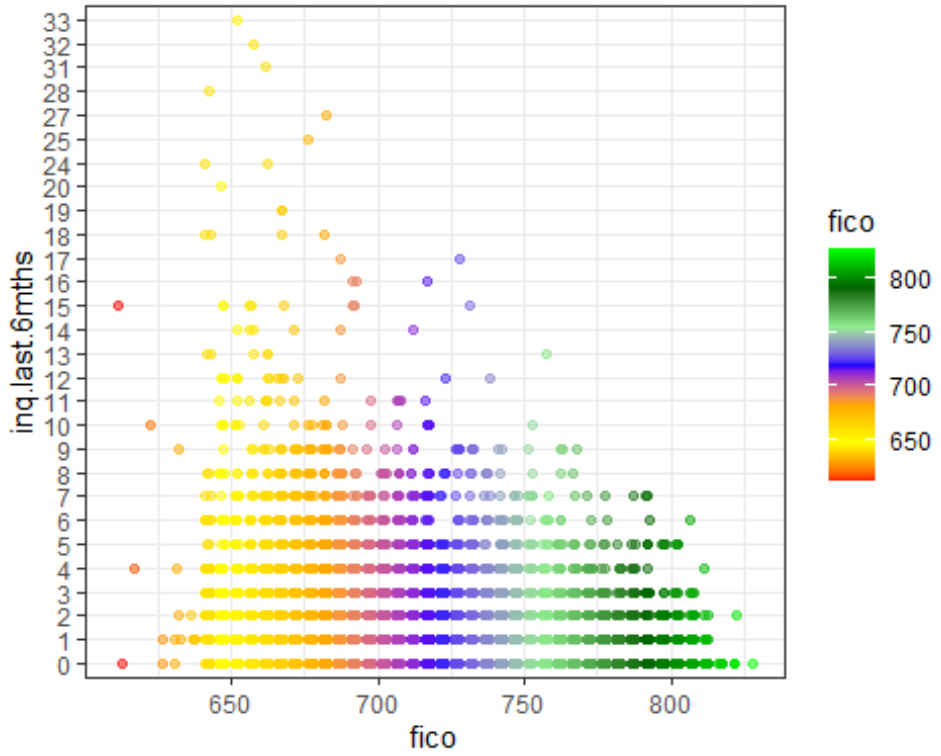

```
#### in the column not.fully.paid 0 means False meaning paid and 1 means true
meaning not paid
#### in short 0 paid and 1 unpaid

ggplot(loan,aes(factor(purpose))) +
geom_bar(aes(fill=not.fully.paid),alpha=0.7) + theme_bw()
```



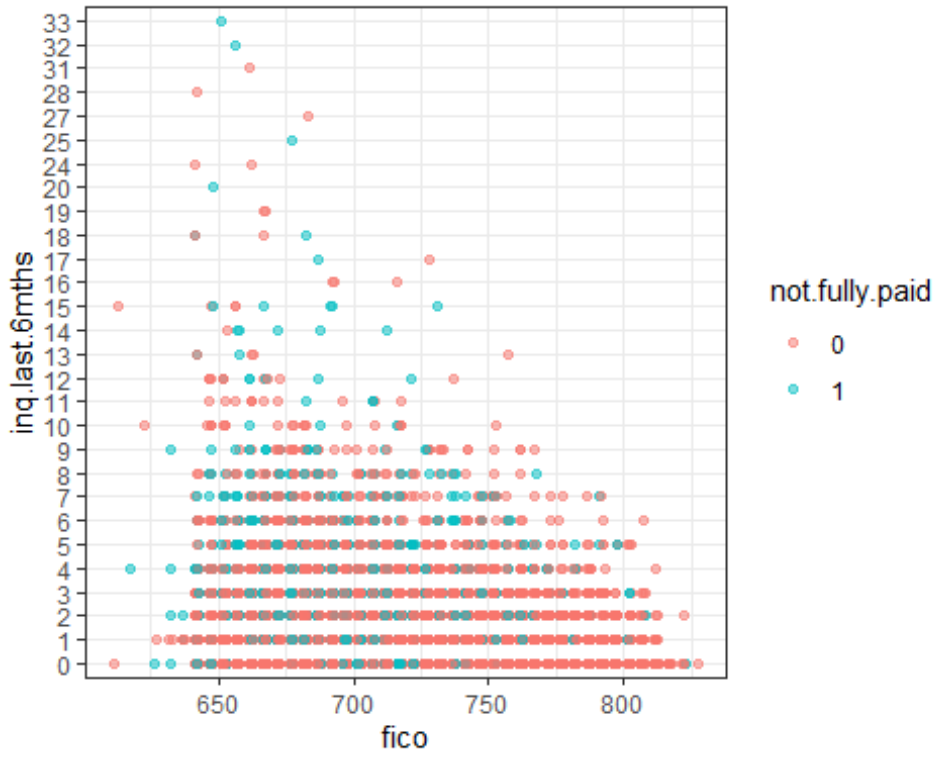

```
ggplot(loan,aes(factor(purpose))) +
geom_bar(aes(fill=not.fully.paid),position = 'dodge',alpha=0.7) + theme_bw()
```



```
ggplot(loan,aes(fico,inq.last.6mths)) +
geom_point(position=position_jitter(w=1, h=0),aes(color=fico),alpha=0.5) +
scale_color_gradientn(colours = c('red','yellow','orange','blue','light
green','dark green','green')) + theme_bw()
```




```
ggplot(loan,aes(fico,inq.last.6mths)) +
geom_point(position=position_jitter(w=1,
h=0),aes(color=not.fully.paid),alpha=0.5) + theme_bw()
```




```
#### train and test sample

sample <- sample.split(loan$not.fully.paid,SplitRatio = 0.7)

train <- subset(loan,sample == TRUE)
test <- subset(loan,sample == FALSE)

#### model building

initial.model <- svm(not.fully.paid ~.,train)

summary(initial.model)

##
## Call:
## svm(formula = not.fully.paid ~ ., data = train)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  1
##
## Number of Support Vectors:  2837
##
##  ( 1764 1073 )
```

```
##
##
## Number of Classes:  2
##
## Levels:
##  0 1
```

```
first.predict <- predict(initial.model,test)
```

```
table(first.predict,test$not.fully.paid)
```

```
##
## first.predict    0    1
##             0 2413  460
##             1    0    0
```

```
#### its not really effective like this because we did not choose the gamma
nor the cost so now lets make the final model with those in mind
#### we can achieve better gamma and cost with using tune and then later we
can use that cost and gamma inside the final model to make it more optimum
```

```
tune.results <- tune(svm,train.x=not.fully.paid~., data=train,kernel='radial',
                     ranges=list(cost=c(1,10), gamma=c(0.1,1)))
```

```
#### we could have done more complicated tuning with more variables inside
the cost and gamma but it needs a faster computer which i dont have
#### so we will settle for this summary
#### best parameters are cost = 2 and gamma = 0.1
```

```
final.model <- svm(not.fully.paid ~.,train,kernal='radial',cost=2,gamma=0.1)
```

```
#### predict now
```

```
predict.model <- predict(final.model,test)
```

```
table(predict.model,test$not.fully.paid)
```

```
##
## predict.model    0    1
##             0 2412  459
##             1    1    1
```

```
Acc.model <- (2410+3)/(2410+3+457+3)
```

```
print(Acc.model)
```

```
## [1] 0.8398886
```

```
#### well it seems like its not the best model but for the better model we
will need better computer
#### as we will have to put more values inside the cost and gamma of the
tuning procedure
```