R Notebook - Loan Data Project - Support Vector Machine - Bryan Honeck

Code

This data comes from LendingClub.com where we will be trying to classify/predict whether or not the person borrowing money paid back the loan entirely. We will use the support vector machine function in R to build the model and eventually make these predictions.

Hide

```
loan_data <- read.csv("C:/Users/Bryan/Desktop/Machine Learning Projects/practice/loan_data.csv")

head(loan_data)
```

	credit.policy <int>	purpose <fctr>	int.rate <dbl>	installment <dbl>	log.annual.inc <dbl>	dti <dbl>	fico <int>
1	1	debt_consolidation	0.1189	829.10	11.35041	19.48	737
2	1	credit_card	0.1071	228.22	11.08214	14.29	707
3	1	debt_consolidation	0.1357	366.86	10.37349	11.63	682
4	1	debt_consolidation	0.1008	162.34	11.35041	8.10	712
5	1	credit_card	0.1426	102.92	11.29973	14.97	667
6	1	credit_card	0.0788	125.13	11.90497	16.98	727

6 rows | 1-8 of 14 columns

Hide

```
str(loan_data)
```

```
'data.frame':   9578 obs. of  14 variables:
 $ credit.policy    : int  1 1 1 1 1 1 1 1 1 1 ...
 $ purpose          : Factor w/ 7 levels "all_other","credit_card",..: 3 2 3 3 2 2 3 1 5 3 ...
 $ 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 ...
```

Hide

```
summary(loan_data)
```

```
 credit.policy                 purpose        int.rate       installment     log.annual.inc
dti
 Min.   :0.000   all_other         :2331   Min.   :0.0600   Min.   : 15.67   Min.   : 7.548   Mi
n.   : 0.000
 1st Qu.:1.000   credit_card       :1262   1st Qu.:0.1039   1st Qu.:163.77   1st Qu.:10.558   1s
t Qu.: 7.213
 Median :1.000   debt_consolidation:3957   Median :0.1221   Median :268.95   Median :10.929   Me
dian :12.665
 Mean   :0.805   educational       : 343   Mean   :0.1226   Mean   :319.09   Mean   :10.932   Me
an   :12.607
 3rd Qu.:1.000   home_improvement  : 629   3rd Qu.:0.1407   3rd Qu.:432.76   3rd Qu.:11.291   3r
d Qu.:17.950
 Max.   :1.000   major_purchase    : 437   Max.   :0.2164   Max.   :940.14   Max.   :14.528   Ma
x.   :29.960
                 small_business    : 619
      fico       days.with.cr.line   revol.bal         revol.util    inq.last.6mths    delinq.2y
rs
 Min.   :612.0   Min.   :  179     Min.   :      0   Min.   :  0.0   Min.   : 0.000   Min.   :
0.0000
 1st Qu.:682.0   1st Qu.: 2820     1st Qu.:   3187   1st Qu.: 22.6   1st Qu.: 0.000   1st Qu.:
0.0000
 Median :707.0   Median : 4140     Median :   8596   Median : 46.3   Median : 1.000   Median :
0.0000
 Mean   :710.8   Mean   : 4561     Mean   :  16914   Mean   : 46.8   Mean   : 1.577   Mean   :
0.1637
 3rd Qu.:737.0   3rd Qu.: 5730     3rd Qu.:  18250   3rd Qu.: 70.9   3rd Qu.: 2.000   3rd Qu.:
0.0000
 Max.   :827.0   Max.   :17640     Max.   :1207359   Max.   :119.0   Max.   :33.000   Max.   :1
3.0000

    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
```

Some of these attributes are categorical; let's convert them using factor()

Hide

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

Hide

```
str(loan_data)
```

```
'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          : Factor w/ 7 levels "all_other","credit_card",..: 3 2 3 3 2 2 3 1 5 3 ...
 $ 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 ...
```

Hide

```
summary(loan_data)
```

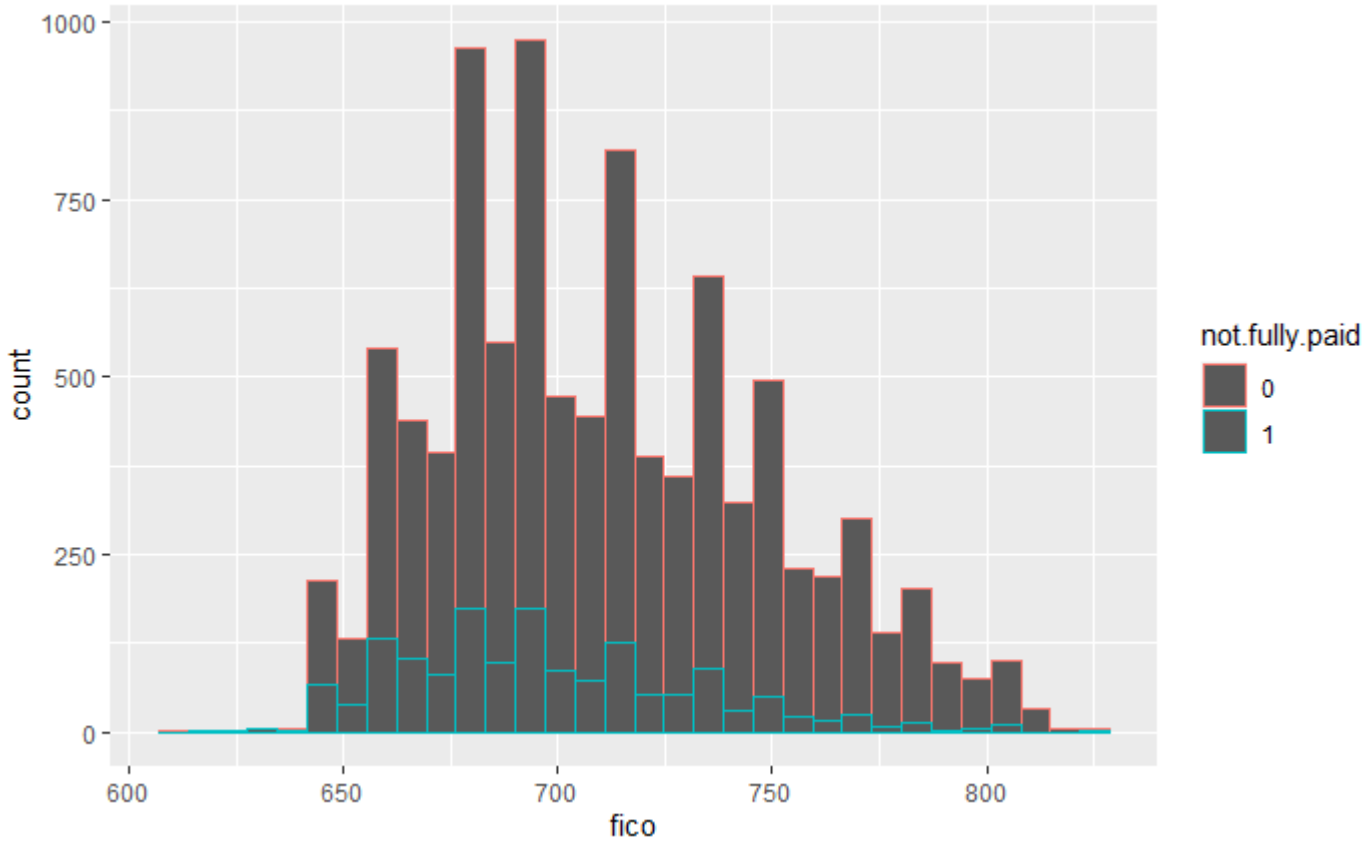
```
 credit.policy              purpose        int.rate       installment     log.annual.inc
dti
 0:1868        all_other         :2331   Min.   :0.0600   Min.   : 15.67   Min.   : 7.548   Min.
: 0.000
 1:7710        credit_card       :1262   1st Qu.:0.1039   1st Qu.:163.77   1st Qu.:10.558   1st
Qu.: 7.213
               debt_consolidation:3957   Median :0.1221   Median :268.95   Median :10.929   Medi
an :12.665
               educational       : 343   Mean   :0.1226   Mean   :319.09   Mean   :10.932   Mean
:12.607
               home_improvement  : 629   3rd Qu.:0.1407   3rd Qu.:432.76   3rd Qu.:11.291   3rd
Qu.:17.950
               major_purchase    : 437   Max.   :0.2164   Max.   :940.14   Max.   :14.528   Max.
:29.960
               small_business    : 619
      fico       days.with.cr.line   revol.bal         revol.util    inq.last.6mths  delinq.2yrs
pub.rec
 Min.   :612.0   Min.   :  179     Min.   :      0   Min.   :  0.0   0      :3637   0      :8458
0:9019
 1st Qu.:682.0   1st Qu.: 2820     1st Qu.:   3187   1st Qu.: 22.6   1      :2462   1      : 832
1: 533
 Median :707.0   Median : 4140     Median :   8596   Median : 46.3   2      :1384   2      : 192
2:  19
 Mean   :710.8   Mean   : 4561     Mean   :  16914   Mean   : 46.8   3      : 864   3      :  65
3:   5
 3rd Qu.:737.0   3rd Qu.: 5730     3rd Qu.:  18250   3rd Qu.: 70.9   4      : 475   4      :  19
4:   1
 Max.   :827.0   Max.   :17640     Max.   :1207359   Max.   :119.0   5      : 278   5      :   6
5:   1
                                                                     (Other): 478   (Other):   6
 not.fully.paid
 0:8045
 1:1533
```

Let's do some EDA.

Hide

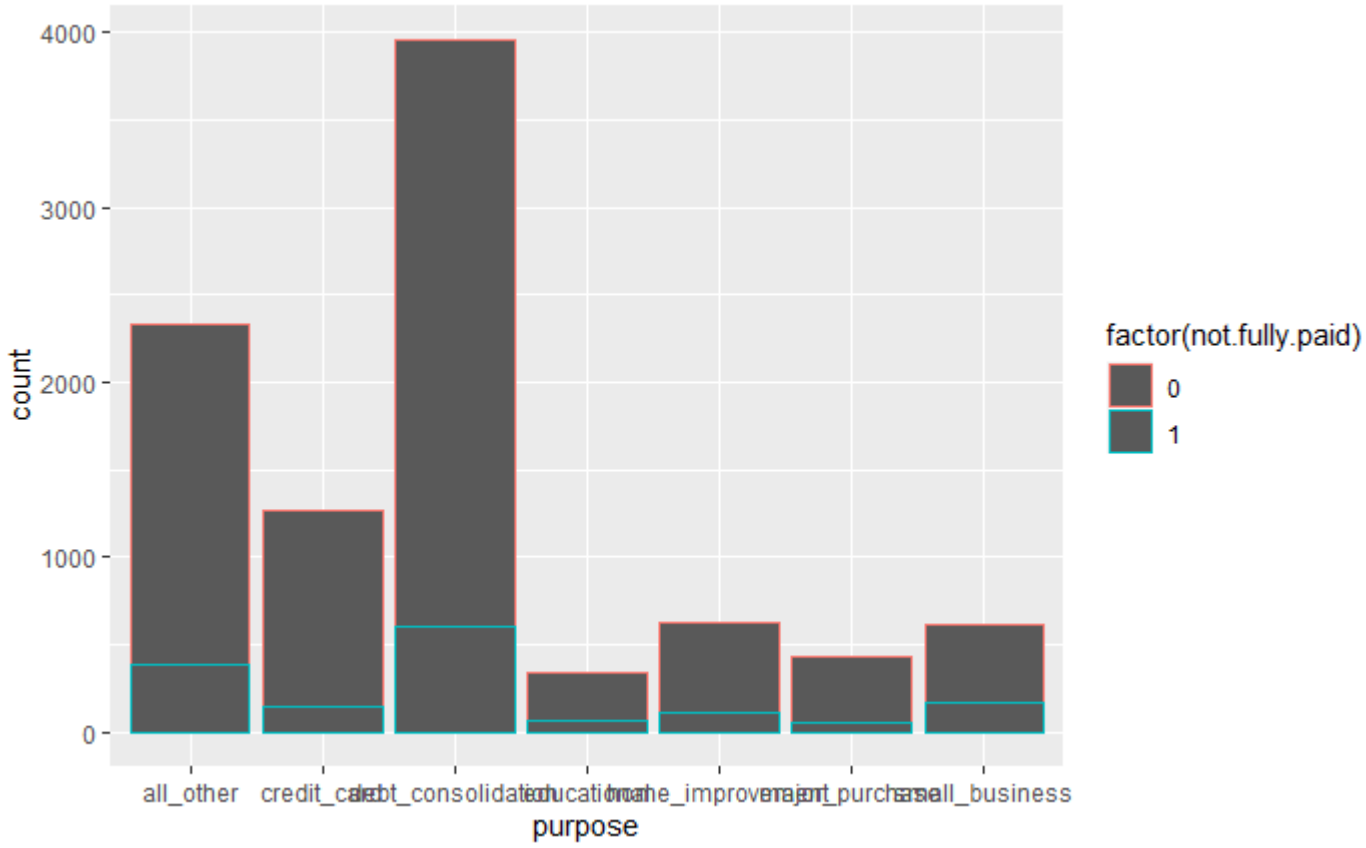
```
library(ggplot2)
library(plotly)

pl <- ggplot(data = loan_data, aes(fico)) + geom_histogram(aes(color=not.fully.paid), bins = 32)
pl
```



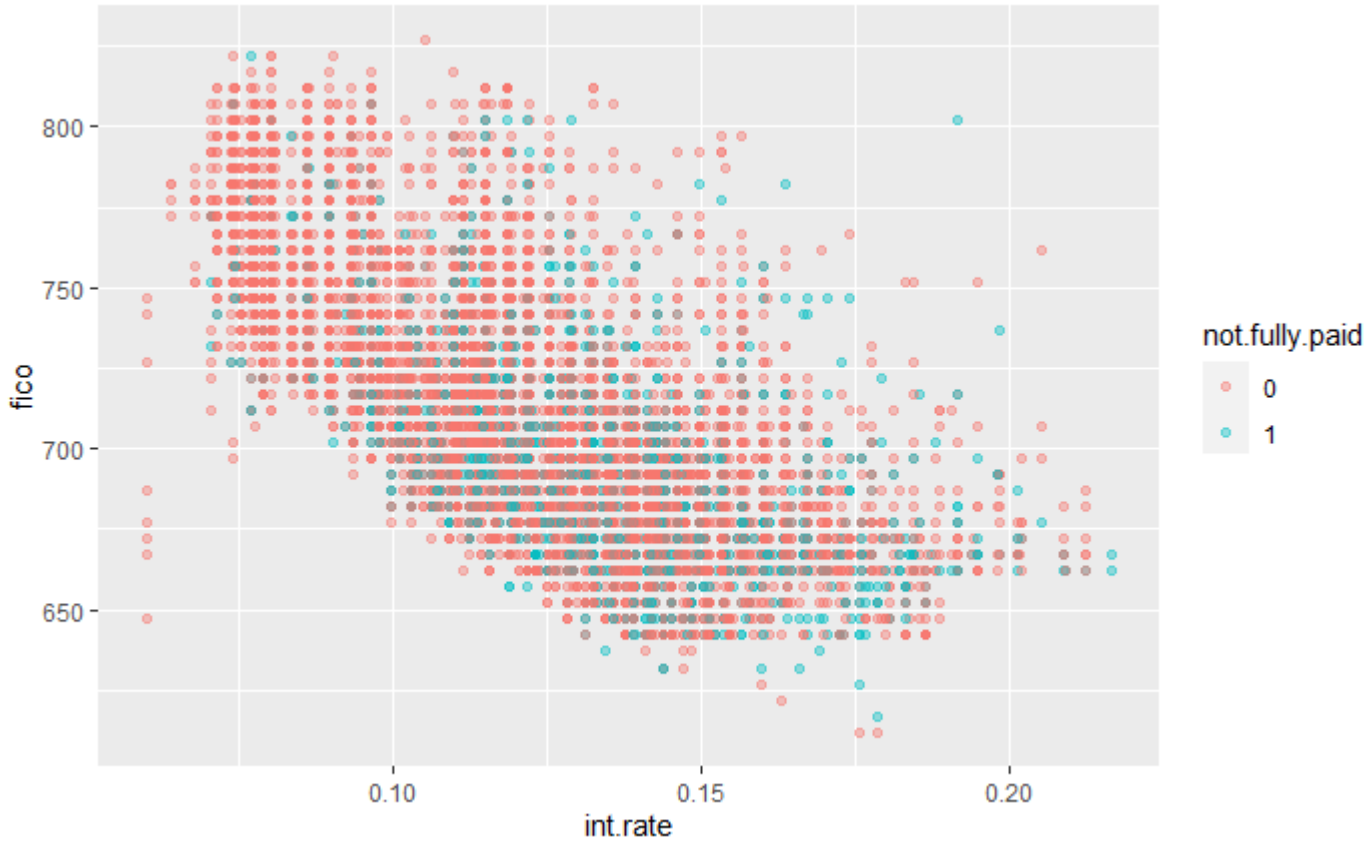
Hide

```
pl2 <- ggplot(data = loan_data, aes(purpose)) + geom_bar(aes(color=factor(not.fully.paid)))
pl2
```



Hide

```
pl3 <- ggplot(data = loan_data, aes(int.rate, fico)) + geom_point(aes(color=not.fully.paid), alp
ha = 0.4)
pl3
```

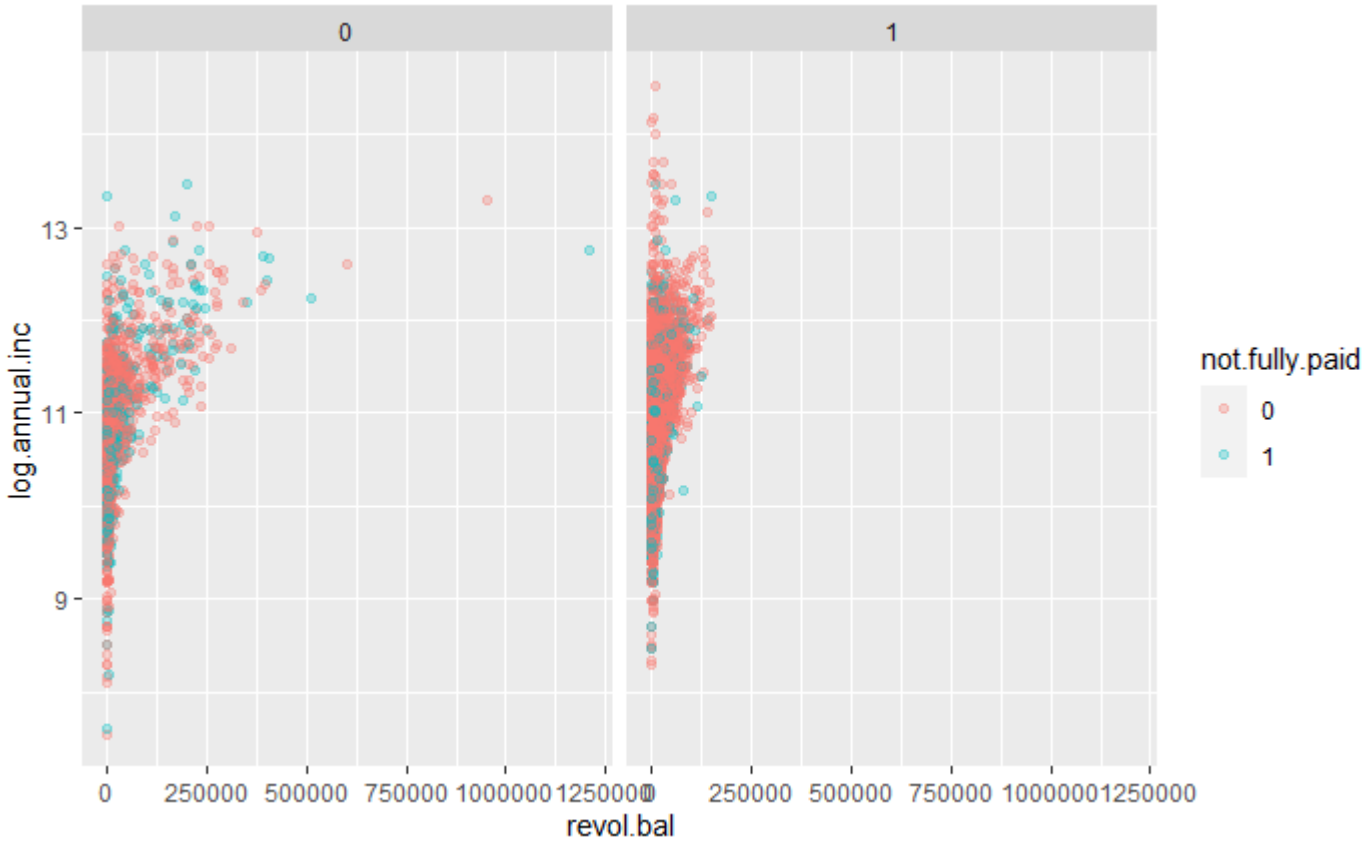


Hide

```
pl4 <- ggplot(data = loan_data, aes(days.with.cr.line, installment)) + geom_point(aes(color=not.fully.paid), alpha = 0.3) + facet_wrap(~credit.policy)
pl4
```

Hide

```
pl5 <- ggplot(data = loan_data, aes(revol.bal, log.annual.inc)) + geom_point(aes(color=not.full
y.paid), alpha = 0.3) + facet_wrap(~credit.policy)
pl5
```



That's a few visualizations to look at. Let's start building our model.

Hide

```
library(caTools)
library(e1071)

sample <- sample.split(loan_data$not.fully.paid, 0.7)

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

Now use svm() function to generate the model.

Hide

```
model <- svm(not.fully.paid ~ ., data = train)
summary(model)
```

```
Call:
svm(formula = not.fully.paid ~ ., data = train)


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

Number of Support Vectors:  4030

 ( 2497 1533 )


Number of Classes:  2

Levels:
 0 1
```

Let's create the predictions:

Hide

```
pred.values <- predict(model, test[1:13])
table(pred.values, test$not.fully.paid)
```

```
pred.values    0    1
          0 8045 1532
          1    0    1
```

These predictions were terrible. All but one were put into one group. The model needs to be tuned.

Hide

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

```
Parameter tuning of ‘svm’:

- sampling method: 10-fold cross validation

- best parameters:
```

cost	gamma
<dbl>	<dbl>
1	0.1

1 row

```
- best performance: 0.1598445

- Detailed performance results:
```

cost <dbl>	gamma <dbl>	error <dbl>	dispersion <dbl>
1	0.1	0.1598445	0.01508116
10	0.1	0.1683014	0.01029372
1	1.0	0.1606797	0.01571739
10	1.0	0.1818741	0.02106618

4 rows

```
NA
```

Now, we run the model with the best cost and gamma from the group we chose.

Hide

```
final.model <- svm(not.fully.paid ~., data = train, cost=10, gamma=0.1)
final.pred.values <- predict(final.model, test[1:13])
table(final.pred.values, test$not.fully.paid)
```

```
final.pred.values    0    1
                0 8037 1100
                1    8  433
```

This is better, but still not perfect. We could continue to improve these values, but the run time can be costly. With just my machine, this can be difficult to accomplish, but with several nodes, it would be reasonable to continue adjusting the vectors.